

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique



Université Dr. Tahar Moulay de Saïda
Faculté de la Technologie
Département d'Electrotechnique



Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme de

Master (LMD)

Spécialité : ELECTROTECHNIQUE INDUSTRIELLE

Filière : ELECTROTECHNIQUE

Intitulé :

**Performances de simulation en temps réel avec RT-LAB
d'un système d'entraînement électrique sur « OP4510 »**

Présenté par :

- **CHETTI Zoubida**
- **ANGAR Halima**

Devant le jury composé de :

Pr. MEZOUAR Abdelkader
Mr. BENYAHIA Mokhtar
Pr. MEKRI Fatiha

Président
Encadreur
Examineur

Soutenu le 20 / 10 / 2021
Promotion 2020-2021

Remerciements

On remercie Dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mr BENYAHIA Mokhtar, un remerciement spécial pour la qualité de son encadrement exceptionnel, pour sa patience, son aide et ses conseils pour élaborer ce travail.

Nos sincères remerciements à tous les membres du jury : Mr. MEZOUAR Abdelkader et Mme. MEKRI Fatiha, qui nous ont fait l'honneur d'avoir accepté d'examiner et évaluer notre modeste travail.

En fin toute personne qui a participé de près ou de loin à l'accomplissement de ce mémoire soit sincèrement remerciée ainsi que les enseignants qui ont partagés avec nous à tous les moments de notre chemin d'étude.

Table des matières

Liste des figures

Liste des tableaux

Notations et abréviations

Introduction générale 1

Chapitre 1 : Simulation en temps réel : principe et différents aspects

1.1 Introduction 4

1.2 Simulation en temps réel 5

1.3 Différents outils de simulation 6

1.3.1 Types de simulateurs 6

1.3.2 Différences entre simulation temps réel et simulation ordinaire 9

1.4 Simulateur OPAL-RT 11

1.5 Conclusion 13

Chapitre 2 : Propriétés de la simulation en temps réel avec OPAL-RTLAB

2.1 Introduction 14

2.2 Simulateurs OPAL-RT 14

2.3 Types de simulateurs développés par OPAL RT-LAB 15

2.3.1 Le simulateur OPAL 4510 16

2.3.2 Caractéristiques spécifiques 16

2.3.3 Architecture et périphériques 16

2.4 Les Avantages de simulation avec RT-LAB	18
2.5 Concepts et apport de simulation en temps réel avec OPAL RT-LAB	19
2.6 Les différentes méthodes de simulation en temps réel	22
2.6.1 Modèle dans la boucle (Model In the Loop: MIL)	23
2.6.2 Prototypage de contrôle rapide (Rapid control prototyping : RCP)	23
2.6.3 Matériel dans la boucle (Hardware In the Loop : HIL)	25
2.6.4 Matériel de puissance dans la boucle (Power Hardware In the Loop : PHIL)	26
2.6.5 Simulations hybrides complexes (Complex hybrid simulations :HYBRID).....	27
2.7 Package logiciel et Outils de simulation avec les simulateurs OPAL-RT	28
2.7.1 HYPERSIM : « Le simulateur des réseaux électriques (de demain) »	29
2.7.1.1 Les composants principaux de HYPERSIM	30
2.7.1.2 Applications typiques.....	30
2.7.2 eMEGASIM.....	31
2.7.3 RT-LAB	32
2.7.3.1 Applications typiques.....	33
2.7.4 eFPGASIM	34
2.7.4.1 eFPGASIM pour les transports électriques.....	35
2.7.4.2 eFPGASIM : boîte à outils logicielle	35
2.7.4.3 Applications typiques.....	37
2.7.5 ePHASORSIM	37
2.8 Conclusion.....	38

Chapitre 3 : Étapes de Simulation en temps réel sous RT-LAB

3.1 Introduction	39
3.2 Structure du modèle	40
3.2.1 Regroupement en sous-système	40
3.2.2 Ajout de blocs OpComm	42
3.2.3 Ajout des blocs d'entrées- sorties	44
3.3 Présentation de l'interface graphique de RT-LAB et la préparation d'un modèle	45
3.4 Construction du modèle et lancement de simulation sous RT-LAB	50
3.4.1 Construction du modèle sous RT-LAB : « Build »	50
3.4.2 Chargé et exécuter le modèle sur la cible	51
3.5 Enregistrement de donnée	53
3.5.1 Enregistrement de données	53
3.5.2 Enregistrement et affichage avec « ScopeView »	54
3.5.2.1 Caractéristiques principales.....	55
3.6 Conclusion.....	55

Chapitre 4 : Applications sur un système d'entraînement électrique d'une MSAP

4.1 Introduction	57
4.2 Description générale du système (circuit)	58
4.3 Entraînement de la machine synchrone à aimants permanents (PMSM drive).....	60
4.3.1 Intérêt de La machine synchrone à aimants permanents « PMSM ».....	60
4.3.1.1 Modèles du moteur (PMSM) disponibles	61

4.3.2 Description du bloc convertisseur à trois niveaux (3level TSB – RTE Drive)	61
4.4 Simulation du modèle « TRAIN TRACTION DRIVE » sur L'OP4510	63
4.4.1 Étapes de déroulement de la simulation en temps réel	65
4.4.2 Résultats de simulation	72
4.4.2.1 Résultats d'enregistrement avec OpwriteFile (Matlab)	72
4.4.2.2 Configuration de l'enregistrement sous ScopeView	73
4.4.2.3 Enregistrement des résultats avec ScopeView	76
4.5 Interprétations des simulations	80
4.6 Conclusion	81
Conclusion Générale	82
Références bibliographiques	84
Annexes	85

Liste des figures

Liste des figures du chapitre 1

Figure (1. 1) : Principe de la simulation numérique	5
Figure (1. 2) : Différents types des simulateurs	7
Figure (1. 3) : Simulation hybride	8
Figure (1. 4) : Les différentes techniques de simulations	9
Figure (1. 5) : Pas de temps dans différents scénarios (E : Entré/S : Sortie)	11
Figure (1. 6) : Simulation hybride réalisée avec OPAL-RT	12

Liste des figures du chapitre 2

Figure (2. 1) : Les deux hôtes (stations) de Simulations avec OPAL RT-LAB	15
Figure (2. 2) : Vue frontale du simulateur OP4510 + Architecture interne	17
Figure (2. 3): Vue d'arrière de l'OP4510	18
Figure (2. 4): Avantages d'application du simulateur RT-LAB	19
Figure (2. 5-a): Concept de Simulation en temps réel	20
Figure (2. 5-b): Concept de Simulation en temps réel	20
Figure (2. 6-a): domaines d'application des Simulateurs OPAL RT-LAB	21
Figure (2. 6-b): domaines d'application des Simulateurs OPAL RT-LAB	22
Figure (2. 7): Méthodes de Simulation en temps réel avec OPAL RT-LAB	22
Figure (2. 8): Modèle de simulation dans la boucle (Model In the Loop :MIL	23

Figure (2. 9- a): technique de Prototypage de contrôle rapide (Rapid control prototyping : RCP)	24
Figure (2. 9- b): Exemple de Développement d'un algorithme de contrôle sur une machine existante	24
Figure (2. 10): mode de simulation HIL: Matériel dans la boucle	25
Figure (2. 11): Exemples de Simulation avec la méthode HIL : Matériel dans la boucle	26
Figure (2. 12): mode de simulation PHIL: simulation : Matériel de puissance dans la boucle	26
Figure (2. 13): Exemples de Simulation avec la méthode PHIL	27
Figure (2. 14): Simulations hybrides complexes : HYBRID	27
Figure (2. 15): Exemple de Simulations hybrides complexes : HYBRID	28
Figure (2. 16): Pack logiciel (SOFT) de simulation temps réel d'OPAL RT-LAB	29
Figure (2. 17): Simulation temps réel des réseaux électriques avec HYPERSIM	30
Figure (2. 18): Simulation temps réel avec eMEGASIM	31
Figure (2. 19): fréquences des phénomènes transitoires simulés par le package OPAL-RT-LAB	33
Figure (2. 20): Simulation avec eFPGASIM sous RT-LAB	34
Figure (2. 21): eHS : Solveur électrique pour FPGA générique et reprogrammable	36
Figure (2. 22): Modèles de convertisseurs modulaires multiniveaux (MMC) sur FPGA	36
Figure (2. 23): structure interne : Co-simulation CPU/FPGA	37

Liste des figures de chapitre 3

Figure (3. 1) : Etapes de simulation sous RT-LAB	39
Figure (3. 2) : les deux principaux sous systèmes que doit contenir un modèle	41

Figure (3. 3) : regroupement des blocs en sous systèmes	41
Figure (3. 4) : Distinction de l'interface de calcul et l'interface graphique	42
Figure (3. 5) : Regroupement des blocs et contenu de chaque sous systèmes	42
Figure (3. 6) : Ajout de l'OpComm de la bibliothèque RT-LAB sous Matlab/Simulink	43
Figure (3. 7): OpComm inséré dans les deux sous systèmes	43
Figure (3. 8) : communication entre sous systèmes via l'OpComm (rouge=synchrone, bleu=asynchrone)	44
Figure (3.09) : traitement des « OpComm » et occupation des cœurs de CPU du simulateur et Host PC	44
Figure (3.10) : fenêtre principale après lancement du logiciel RT-LAB	46
Figure (3.11) : Étapes de l'ajout d'un modèle exemple au projet sous RT-IAB	46
Figure (3.12) : fenêtres de l'ajout d'un modèle au projet sous RT-LAB	47
Figure (3.13): Création d'un nouveau modèle vierge au projet sous RT-LAB	47
Figure (3.14) : fenêtres pour « Add » ou « Import » des modèles existants	48
Figure (3.15): «Edit» et sélection de la version Matlab/Simulink pour ouvrir le modèle	49
Figure (3.16): fenêtres d'activation de la cible « target » PC-Simulateur	50
Figure (3.17) : Construction de modèle sous RT-LAB « Build»	51
Figure (3.18) : Exécution du modèle et console générée par RT-LAB sous Matlab/Simulink	52
Figure (3.19) : La console Simulink des résultats enregistrés générée après reinitialisation	52
Figure (3.20) : Insertion et paramétrage du bloc « OpWriteFile »	54
Figure (3.21) : Principe d'enregistrement de données avec « OpWriteFile »	54

Liste des figures de chapitre 4

Figure (4. 1) : Circuit des différents éléments du modèle étudié	58
Figure (4. 2) : Circuit électrique équivalent du convertisseur TSB 3-Niveaux – RTE-DRIVE	62
Figure (4. 3) : Analyse du cycle d'utilisation pour la technique d'horodatage (TSB) RTE-DRIVE	63
Figure (4. 4) : Simulateur OP4510 relié au PC (du laboratoire LGE)	65
Figure (4. 5-a) : Edition du modèle étudié avec ses sous systèmes	65
Figure (4.5-b) : Les deux sous systèmes du modèle édité	66
Figure (4. 6) : La construction et la compilation du modèle	66
Figure (4.7) : L'assignation et l'activation du mode XHP	67
Figure (4. 8) : Chargement « load » du modèle au simulateur	68
Figure (4. 9) : La console d'affichage générée après chargement du modèle	68
Figure (4.10): phase d'Exécution du modèle sur le simulateur en temps réel	69
Figure (4.11) : réinitialisation de la simulation « reset »	70
Figure (4.12-a) : données de simulation en temps réel avec activation du mode XHP»	70
Figure (4.12-b): données de simulation en temps réel avec (XHP activé) pour les deux sous systèmes	71
Figure (4.13) : données de simulation en temps réel sans mode XHP	71
Figure (4.14) : Allures des couples (ΔT_{ref} , T_m) et vitesses (ω_{ref} , ω_m)	72
Figure (4.15) : Tensions triphasées du moteur	72
Figure (4.16) Courants triphasés du moteur	73
Figure (4.17) : Sélections et choix des signaux à enregistrer pour chaque sous système	73

Figure (4.18) : Les groupes de signaux développés après exécution et enregistrement	74
Figure (4.19) : Lancement de ScopeView	74
Figure (4.20) : l'interface d'affichage (graphique) de ScopeView	75
Figure (4.21-a) : Tensions triphasées du moteur – (enregistrement SV)	76
Figure (4.21-b) : Zoom sur les tensions triphasées du moteur – (enregistrement SV)	76
Figure (4.22) : Courants triphasés du moteur (enregistrement SV)	77
Figure (4.23) : Allures des couples (ΔT_{ref} , T_m) enregistrement SV)	77
Figure (4.24) : Allures des vitesses (ω_{ref} , ω_m) (enregistrement SV)	78
Figure (4.25) : Tensions aux secondaires du transformateur (V_A , V_Y) – (enregistrement SV)	78
Figure (4.26) : Coutants aux secondaires du transformateur (Ia_A , Ia_Y) – (enregistrement SV)	78
Figure (4.27) : Tensions du condensateur coté DC- (enregistrement SV)	79
Figure (4.28) : Courant de l'inductance I_L (enregistrement SV)	79
Figure (4.29) : Courants du pont onduleur I_{br} (enregistrement SV)	79
Figure (4.30) : Temps de calculs et dépassements (enregistrement SV)	80

Liste des tableaux

Liste des tableaux du chapitre 2

Tableau (2. 1) : Configurations des groupes E/S (I/Os) de l'OP 4510	18
--	----

Liste des tableaux du chapitre 3

Tableau (3. 1) : Types de sous-systèmes RT-LAB/Simulink	40
--	----

Liste des tableaux du chapitre 4

Tableau (4. 1) : Allocation de processeur et performances en temps réel	64
--	----

Notations et Abréviations

Abréviations :

HYPERSIM	Simulateur temps réel de grands réseaux
eMEGASIM	Simulateur temps réel des systèmes électriques et d'électronique de puissance
ePPHASORsim	Simulateur temps réel pour les études de stabilité électromécanique
eFPGAsim	Simulateur temps réel d'électronique de puissance sur FPGA
ARTEMiS	Advanced Real-Time Electro-Magnetic Solvers
FPGA	Field Programmable Gate Arrays (un circuit intégré composé d'un réseau de cellules programmables).
eHS	Transmission Control Protocol (Protocol de contrôle de Transmission)
I/O	Input/Output (entrée/sortie).
PC	Ordinateur Personnel
RT	Temps réel
MIL	Modèle dans la boucle
HIL	Matériel dans la boucle
RCP	Prototypage de control .
PHIL	Matériel d'alimentation dans la boucle .
HYPRID	Simulation Hybrides dans la boucle.
TSB	Time stamped bridge
XHP	Extra High Performance
RT-LAB	Real Time Laboratory
SM	Sous-système de calcul
SS	Sous-systèmes de calcul additionnels
SC	Sous-système de l'interface console

SW	Software
HW	Hardware
SV	ScopeView.
PWM	Pulse Width Modulation
MLI	Modulation De Largeur D'impulsion
NPC	Neutral point clamped
SSN	State Space Nodal
GTO	GateTurn-Off thyristor
IGBT	Insulate Gate Bipolar Transistor
PMSM	Permanent Magnet Synchronous Motor.
MSAP	Moteur synchrone à aimants permanents

Symboles :

ω_{ref}	Vitesse de référence.
ω_m	Vitesse du moteur
T_m	Couple moteur (de charge)
T_e	Couple électrique du Moteur
T_{ref}	Couple de référence
ΔT_{ref}	Couple de référence additionnel (charge supplémentaire)
$U_{a,b,c}$	Tensions triphasées du moteur
$I_{a,b,c}$	Courants de la charge (Courants du moteur)
$V_{a,b,c}$	Tensions simples délivrées par le convertisseur
I_L	Courant de l'inductance
V_{Δ}	Tension de l'enroulement D (A)
V_Y	Tension de l'enroulement Y (A)
$I_{a\Delta}$	Courant de l'enroulement D (A)
I_{aY}	Courant de l'enroulement Y (A)
I_{br}	Courant du pont AC/CC

INTRODUCTION GENERALE

INTRODUCTION GÉNÉRALE

Des simulateurs en temps réel ont été développés et utilisés pour le prototypage, l'analyse et la validation d'algorithmes de contrôle. Ces simulateurs offrent des avantages significatifs pour les tests d'équipement physique et la simulation en temps réel des différents appareils. Les chercheurs ont tendance à développer leurs modèles en temps réel à un stade précoce du processus de conception.

Les systèmes électriques sont de plus en plus présents ces dernières décennies. Les entraînements électriques à vitesse variable sont aujourd'hui présents dans toutes les installations industrielles. Le besoin de conserver les ressources naturelles et la croissance du coût de l'énergie ont favorisé l'utilisation de tels types d'entraînements. Il est clair que de ce point de vue, la technique de la vitesse variable peut apporter une contribution très importante.

OPAL-RT est une société spécialisée dans la fabrication des systèmes de simulation en temps réel.

RT-LAB, produit phare D'OPAL -RT, est un logiciel de simulation temps réel qui conjugue performance et convivialité. Depuis sa première mise en œuvre il ya vingt ans pour le bras canadien de L'agence spatiale canadienne, RT-LAB a révolutionné le monde de l'ingénierie des systèmes, que ce soit dans le secteur spatial, terrestre ou naval. Il permet aux ingénieurs et scientifiques d'accélérer le développement de prototypes et de tester rigoureusement de nouvelles technologies et systèmes innovants. Entièrement intégré avec Matlab/simulink, RT-LAB permet la conception de modèles élaborés en simulant leur interaction avec le monde réel par sa souplesse et son extensibilité, il prend en charge les applications de simulation temps réel les plus avancées dans les secteurs de l'automobile, l'aérospatiale, l'électronique de puissance l'entraînement électrique et les réseaux électriques.

C'est est un produit de haute performance distribué à l'échelle mondiale par la société Opal-RT. Il est devenu incontournable dans le monde de la simulation en temps réel et du test avec matériel dans la boucle (**Hardware In the Loop**). Il aide les chercheurs à développer, à tester

et à valider plus rapidement les systèmes d'électronique de puissance, électromécaniques et électriques. Le pack logiciel permet à son utilisateur de développer rapidement des modèles pouvant être simulés en temps réel tout en assurant une plus grande flexibilité lors du processus de la conception des applications industrielles, et en minimisant l'investissement initial, les coûts et les erreurs possibles.

RT-LAB nous propose une série professionnelle de solutions complètes de simulation en temps réel : phénomènes électromagnétiques rapides à la stabilité transitoire des gros systèmes électriques. Comme il permet en génie électrique de travailler à optimiser la sécurité, l'efficacité et les performances des micro-réseaux, des sources d'énergie renouvelables et des grands réseaux électriques interconnectés.

En génie électrique les solutions OPAL-RT répondent à tous les besoins de simulation rapide ou en temps réel des réseaux électriques traditionnels et des systèmes d'électronique de puissance. Les plateformes OPAL-RT aident à mener des études de faisabilité, à développer de nouveaux prototypes, à concevoir et à tester des contrôleurs pour tous types d'applications, dont les petits convertisseurs de puissance, les entraînements électriques hybrides, les grands réseaux électriques et les systèmes intégrant des énergies renouvelables.

Les outils de simulation RT-LAB sont compatibles à Matlab/ simulink avec une intégration parfaite de ses produits (outils) et offrent des solutions logicielles utilisables avec toutes les plates formes de simulateurs OPAL-RT à citer l'OP4510.

Le présent travail a été réalisé sur le simulateur OP4510 au niveau du laboratoire LGE, il fait suite à des travaux préliminaires et des essais sur des modèles installés dans la bibliothèque RT-LAB (boîte d'outils Artemis - drive), il consiste à réaliser des simulations en temps réel avec RT-LAB sur le simulateur, dans le but de comprendre les principes de base et de bien maîtriser les différentes manipulations.

À noter qu'on c'est limité dans ce document à la présentation des étapes de configuration, les pré-réglages, la construction et le développement, ceci est du en raison de la licence limitée, selon les fonctions et outils activés sur une licence temporaire.

Notre mémoire est composé de quatre chapitres :

Le premier chapitre est dédié à la simulation en temps réel, ses avantages, son apport technologique, ses applications et les différents simulateurs utilisés.

Dans le second chapitre, nous faisons une présentation des concepts de simulation en temps réel avec OPAL RT-LAB, les structures, l'architecture interne et externe du simulateur

OP4510. En suite on expose les méthodes, les outils de simulation et le package logiciel de RT-LAB.

Dans le cadre du troisième chapitre on présente les principales étapes à suivre pour exécuter la simulation en temps réel sur RT-LAB avec les configurations et les préréglages nécessaires.

Le quatrième chapitre est consacré à la réalisation d'une simulation en temps réel avec explication du déroulement des étapes de simulation sur l'OP4510. L'application contient un exemple complet sur l'entraînement d'un moteur synchrone à aimants permanents pour la traction électrique d'un train.

On clôture ce mémoire par une conclusion générale et des annexes viendront approfondir certains aspects.

CHAPITRE 1

SIMULATION EN TEMPS REEL :

PRINCIPES ET DIFFERENTS ASPECTS

1.1 Introduction

Le terme simulation rassemble un ensemble de techniques et de procédures qui permettent de reproduire le comportement d'un système physique réel afin d'étudier le dit système dans diverses conditions et sous diverses contraintes. Cette étape arrive suite à l'étape de modélisation qui a pour but de réaliser une représentation mathématique aussi fidèle que possible du système à étudier. La simulation, qui permet d'affiner ce modèle, et de tester des contrôleurs, est de plus en plus répandue dans plusieurs domaines comme l'aéronautique, le domaine militaire ou encore le domaine énergétique. Elle devient une étape indispensable pour la conception de nouveaux systèmes.

Les techniques de simulation modernes offrent des avantages indiscutables sur les vraies expériences appliquées aux systèmes réels. En effet, les expériences sur des systèmes réels peuvent être tout simplement impossibles, comme dans le cas où l'échelle du temps du système est très différente de l'échelle humaine (de l'ordre des nanosecondes ou de l'ordre des milliers d'années).

Les expériences réelles peuvent être aussi dangereuses pour les humains et pour le matériel comme les expériences avec de très grandes tensions qui peuvent engendrer de graves dommages sur le matériel et la mort de l'expérimentateur dans le cas d'un accident. De plus, il est difficile de changer les paramètres de fonctionnement pour une expérience réelle, ce qui limite les possibilités de variation des expériences qui, d'ailleurs, peuvent être très coûteuses. Nous pouvons tirer quatre avantages principaux de l'utilisation des simulations :

- Facilité de manipulation et changement de paramètres ;
- Réduction du temps de la conception des systèmes ;
- Réduction des coûts ;
- Sécurité pour les expérimentateurs et le matériel.

La figure (1.1), montre Principe générale de simulation numérique.

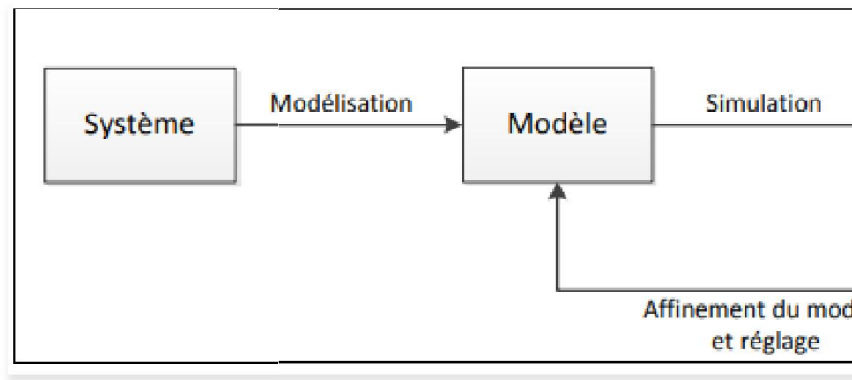


Figure (1.1) : Principe de la simulation numérique

1.2 Simulation en temps réel

La simulation en temps-réel est une technique aujourd'hui largement utilisée par les secteurs de l'industrie de haute technologie tels que l'automatique et l'aéronautique (avion, satellite, commande haute performance, etc..). Pour ces industries, la simulation en temps réel avec intégration de matériel dans la boucle permet de faire le prototypage rapide des systèmes de contrôle de façon économique, sécuritaire et rapide.

La simulation en temps réel est un outil d'analyse puissant, permettant de prévoir le comportement d'un système sous l'action d'un événement particulier et de voir son évolution en temps réel. Cette méthode de conception permet de déceler les problèmes potentiels du système, de réduire le facteur d'erreur humaine et les risques d'interruption du système. L'utilisation du temps réel est donc un outil idéal pour concevoir des systèmes de toute sorte comparativement aux méthodes de conception traditionnelles. [01]

À savoir que la simulation est une technique de résolution de problèmes, la simulation et le test en temps réel nous permettent de :

- Affiner et vérifier le bon fonctionnement des systèmes de contrôle avec le hardware.
- Explorer et tester régulièrement de nouvelles idées avec une plateforme flexible et évolutive.
- Tester le système de contrôle implémenté sur son hardware, même en cas d'indisponibilité du système physique.
- Explorer les scénarios et les interactions hardware s'avérant complexes, onéreux ou dangereux à reproduire sur du hardware de production.
- Éviter les défauts de conception coûteux en détectant les erreurs suffisamment tôt pour que leur correction reste rentable.

1.3. Différents outils de simulation [01], [02]

Avec la progression et la complexification continue des dispositifs et des systèmes électriques et d'électronique de puissance, la nécessité d'utiliser des moyens de simulation de plus en plus pointus, précis et rapides se fait importante et pressante afin de concevoir et de tester efficacement les nouvelles topologies et les méthodes de contrôle appropriées.

La simulation temps réel et la simulation matériel-en-boucle sont reconnues comme deux outils indispensables au design et à la conception de nouveaux systèmes, spécialement les systèmes électriques et d'électronique de puissance. Ces systèmes ont des contraintes de fonctionnement difficiles et exigeantes du genre

- période d'échantillonnage très faible;
- vitesse d'acquisition de données très élevée;
- vitesse de communication très élevée;
- une très grande puissance de calcul.

Pour ces raisons, le design de ces systèmes, demande des outils de simulation très fiables, très pointus technologiquement et souvent très chers, pouvant répondre aux contraintes citées ci-haut.

Les constructeurs de simulateurs temps réel ne sont pas nombreux dans le monde, parmi les systèmes reconnus pour leur fiabilité et qui sont très utilisés nous pouvons citer :

- RTDS (Real Time Digital Simulator) : Simulateur temps réel développé dans le Manitoba, Canada dans les années 1980 au centre de recherche Manitoba MHI (Manitoba Hydro International). C'est un simulateur temps réel numérique spécialisé dans la simulation des réseaux électriques à hautes tensions HVDC et HVAC.
- Hypersim : Simulateur temps réel numérique développé par Hydro Québec depuis 25 ans. Il est spécialisé dans la simulation des réseaux électriques et des systèmes d'électronique de puissance.

1.3.1 Types de simulateurs

Les simulateurs peuvent être classés en deux principaux groupes : Les simulateurs en temps différé et les simulateurs temps réel. Cette dernière catégorie est divisée en simulateurs analogiques, simulateurs numériques et simulateurs hybrides, comme il est illustré par la figure (1.2).

Les simulateurs hors temps réel sont du type numérique, ils sont des simulateurs d'étude et ont pour but de développer et de concevoir de nouveaux systèmes. Ils permettent d'implémenter

une modélisation d'un système bien donné et de la simuler (souvent en assemblant des blocs présentant les éléments basiques du système, ces éléments contiennent les équations et les constantes de leur fonctionnement, et ils offrent la possibilité de configurer d'autres paramètres qui définissent le mode de fonctionnement).

Les logiciels de simulation sont nombreux et très variés, nous pouvons en citer les logiciels suivants parmi les plus connus : EMTP, Matlab/Simulink, COMSOL, etc. Ces logiciels simulateurs sont souples et très flexibles, et ils permettent de simuler et de réguler des systèmes très variés.

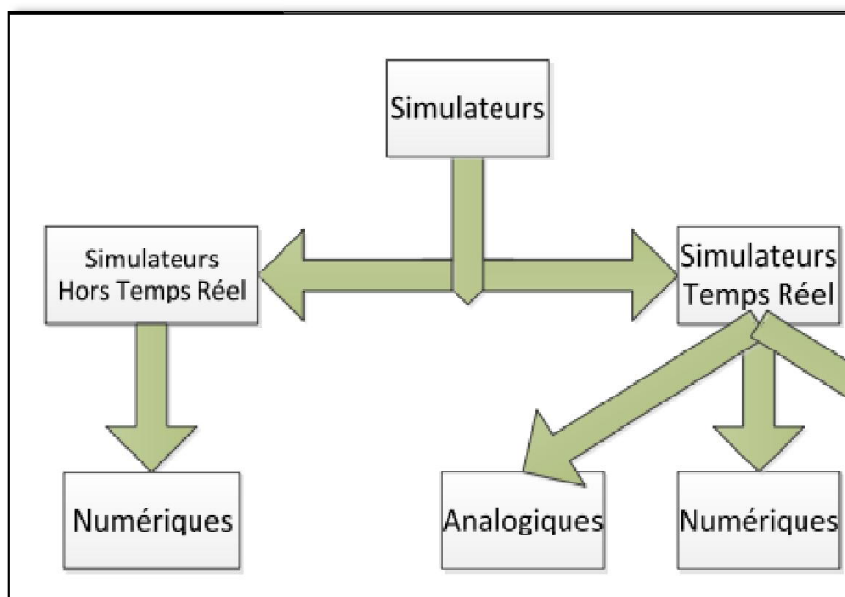


Figure (1.2) : Différents types des simulateurs

L'autre classe de simulateurs, soit les simulateurs temps réel, peut être divisée sur trois sous-classes :

- **Simulateurs temps réel analogiques** : La première génération de simulateurs, c'est une représentation physique réduite du vrai système physique, par exemple, un réseau électrique peut être simulé en utilisant un micro réseau avec des vraies composantes électriques à dimensionnement réduit. Ces simulateurs ont l'avantage de fonctionner naturellement en temps réel, mais ils ont beaucoup plus d'inconvénients comme la perte de précision (en réduisant la taille), le manque de flexibilité pour changer la configuration ou la topologie utilisée, ou encore les coûts élevés.
- **Simulateurs temps réel numériques** : Ce type de simulateurs est le plus utilisé, leur émergence est due principalement à l'augmentation de la puissance de calcul des ordinateurs. Ils ont beaucoup d'avantages comme la grande souplesse et la flexibilité à

changer les modèles et à varier les conditions de test, la réduction des coûts et la sécurité du personnel et de l'équipement.

▪ **Simulateurs temps réel hybrides** (Matériel en boucle / Hardware-In-the-Loop) :

Ce genre de simulateurs est une combinaison des simulateurs numériques et analogiques car il comporte deux parties, une partie analogique représentée par une partie du système réel et une deuxième partie numérique qui contient le modèle numérique de ce qui reste du système. Prenant en exemple le cas de l'électronique de puissance, il devient possible de simuler un contrôleur avec un vrai système ou, au contraire, brancher un vrai contrôleur aux bornes d'un simulateur comportant un modèle d'un circuit de puissance. Ce type de simulateurs a l'avantage d'avoir la flexibilité et la sécurité des simulateurs numériques et le réalisme des simulateurs analogiques. La figure (1.3) montre les différentes combinaisons que peut offrir un simulateur hybride.

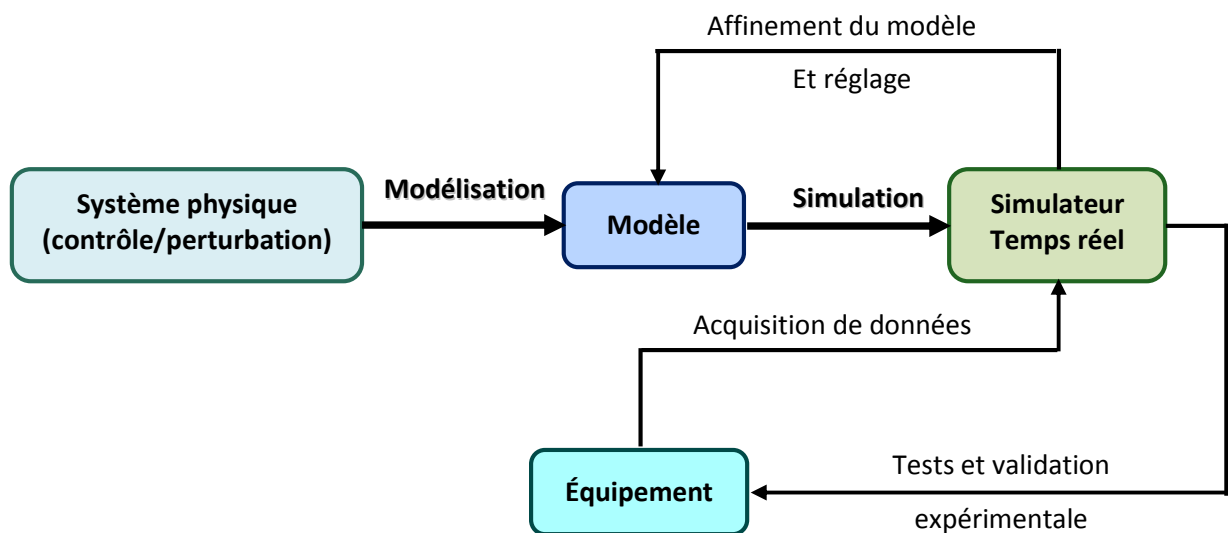


Figure (1.3) : Simulation hybride

Dans la figure (1.4), sont représentés différents schémas. Le premier schéma représente une expérimentation réelle entre un système et un contrôleur. Les deuxième et troisième schémas représentent des simulations purement numériques avec une différence pour le troisième modèle consistant à passer les signaux physiquement (par des fils électriques) entre le modèle du système et le modèle du contrôleur. La quatrième figure est celle d'une simulation hybride avec un modèle du système et un contrôleur réel. Enfin, le dernier est aussi une simulation hybride mais avec un système réel et un modèle du contrôleur.

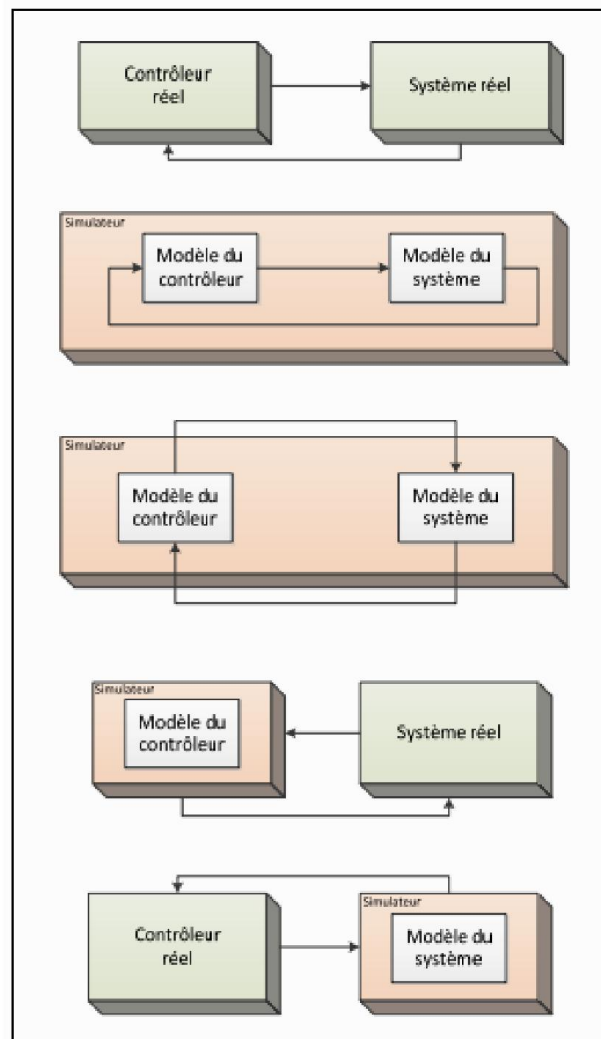


Figure (1.4) : Les différentes techniques de simulations

1.3.2 Différences entre simulation temps réel et simulation ordinaire

Ce qui caractérise le plus la différence entre la simulation temps réel et la simulation hors temps réel est la notion de temps. Pour une simulation nous pouvons définir trois types de temps à prendre en considération [01]:

- Temps physique : Le temps que prend le phénomène physique ou le système conçu pour s'effectuer ;
- Temps simulé : C'est une reproduction du temps physique qui peut être mise à une échelle différente, soit plus grande soit plus petite ;
- Temps absolu : Le temps réel qui s'écoule pendant le déroulement de la simulation.

Une simulation temps réel est une simulation dans laquelle le temps physique, le temps simulé et le temps absolu sont synchronisés, cela veut dire qu'une simulation prendrait exactement le même temps que le phénomène réel pour être réalisée.

Un autre aspect très important qui fait la différence entre une simulation temps réel et une simulation hors temps-réel est le pas de temps. Tandis qu'il peut être variable pour une simulation hors temps réel, il doit être impérativement fixe pour une simulation temps réel. Le pas de temps, ou pas de calcul, est le laps de temps entre deux impulsions d'horloge. Ce laps de temps doit permettre de réaliser les opérations suivantes :

- ✓ Acquisition des données à partir de l'équipement à l'aide des cartes E/S ;
- ✓ Résoudre les équations ;
- ✓ Restitution des signaux vers l'équipement ;
- ✓ Dans le cas de multiprocesseur il faut prendre en considération le temps nécessaire pour transporter les données.

Cette notion (pas de temps) est appelée contrainte temps réel. Un pas de calcul qui ne permet pas d'effectuer toutes les étapes précédentes ne répond pas à cette contrainte ce qui enlève la spécificité temps réel de la simulation ; Dans ce cas, la simulation est souvent irréalisable. Dans le cas du simulateur étudié dans le cadre de ce projet, cela n'empêche pas de réaliser la simulation mais le nombre de cas où cette contrainte n'est pas respectée est comptabilisé, le système sera instable et les résultats non concluants.

Le choix du pas de calcul est délicat car un pas de temps très grand diminue la précision des résultats et engendre l'instabilité du système, alors qu'un pas de temps très petit ne permettra pas de respecter la contrainte temps réel. La figure suivante (1.5) montre les différents scénarios possibles : le cas où le pas de temps est parfait car il permet de réaliser toutes les opérations, le cas où le pas est très grand ce qui fait qu'il y a une perte de temps, et enfin le cas où le pas de temps est très petit ce qui ne permet pas de restituer les signaux au bon moment.

Le temps alloué pour l'acquisition et la restitution des données est constant et dépend des cartes E/S utilisées, mais celui alloué au calcul est variable et dépend du modèle utilisé. Pour alléger le pas de temps il faut, donc, minimiser le temps de calcul. Cela peut être fait principalement de deux manières, soit en utilisant le parallélisme et donc diviser la charge de calcul sur plusieurs processeurs, soit en utilisant des méthodes d'intégrations spécifiques aux simulations temps réel qui sont :

- La méthode des trapèzes : méthode très précise dans le cas de signaux avec peu de discontinuités ;
- La méthode d'Euler régressive : moins précise que la méthode des trapèzes mais beaucoup meilleure pour les signaux discontinus ;
- La méthode mixte (Weight-averaged) : un mélange des deux méthodes qui combine les avantages des deux.

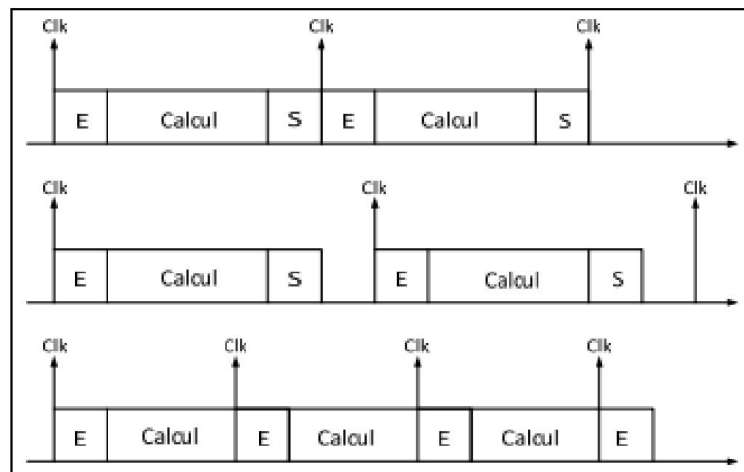


Figure (1.5) : Pas de temps dans différents scénarios (E : Entrée / S : Sortie)

1.4. Simulateur OPAL-RT

Dans le cadre de ce projet, nous avons utilisé un simulateur temps réel pour valider les résultats obtenus. Même si nous ne disposons pas de prototype, la simulation temps réel numérique est plus rigoureuse et plus représentative du comportement réel. Le simulateur utilisé est un simulateur OP4510 développé par la compagnie montréalaise OPAL-RT.

Ce simulateur présente plusieurs avantages; la possibilité de simuler des modèles Matlab/Simulink est l'un de ces points forts, en effet, en apportant quelques modifications et quelques nouveaux blocs un modèle Simulink sera converti et compilé facilement sans réécriture de code. Plusieurs blocs Simulink ont été développés pour permettre l'interfaçage avec la plupart des cartes E/S utilisés. Ce simulateur repose en grande partie sur des technologies existantes telles que TCP/IP, UDP et FireWire pour la communication, ce sont des technologies connues et fiables ce qui assure la flexibilité du système. La puissance de calcul du simulateur est très importante, ce qui permet de simuler des modèles complexes avec de petits pas de temps. [02]

Comme il est illustré à la figure (1.6), le simulateur d'OPAL-RT peut être divisé en deux parties. Une partie appelé Station hôte (Host station), qui contient le logiciel et qui permet de

commander la simulation, d'interagir avec le système simulé et d'afficher les résultats. La deuxième partie est appelée station cible (Target station), qui se résume en un supercalculateur (Jusqu'à 64 cœurs) modulé avec des cartes E/S, numériques et analogiques (Les cartes comportent des modules de conversion analogique-numérique et numérique-analogique et des cartes FPGA). Cette deuxième partie contiendra le modèle à simuler et pourra être connecté à un prototype physique. Les stations cibles de calcul fonctionnent sous les systèmes d'exploitation temps réel QNX ou RedHat.

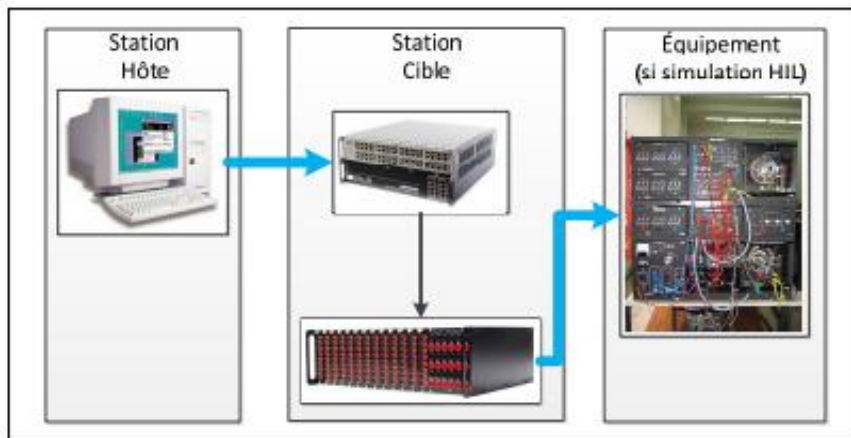


Figure (1.6) : Simulation hybride réalisée avec OPAL-RT

En plus de l'équipement hardware cité ci-haut, OPAL-RT dispose d'une suite logicielle qui permet de simplifier et d'optimiser la simulation temps réel. Nous allons exposer les applications les plus utiles : [02]

- Logiciel RT-Lab : Logiciel qui sert d'interface entre les stations hôtes et les stations cibles. Il permet d'ouvrir une instance du logiciel Matlab/Simulink afin d'éditer un modèle. Le modèle étant validé, le logiciel permet de compiler et de charger le code dans la station cible (Calcul et connexion avec l'équipement réel). Au cours de la simulation, le logiciel permet de visualiser les résultats et d'intervenir pour changer les paramètres de fonctionnement du modèle.
- ARTEMIS : Un ensemble de blocs Matlab/Simulink qui contiennent des méthodes et des algorithmes permettant d'améliorer la précision des méthodes d'intégrations utilisés dans Sim Power System. En effet, la méthode d'intégration utilisée dans Sim Power System est la méthode Tustin (Méthode des trapèzes), elle n'a pas été conçue pour les simulations temps réel et n'est pas très précise en cas de signaux discontinus et très oscillatoires. Pour pouvoir utiliser directement les blocs de Sim Power System, ce qui est un des avantages les plus importants d'OPAL-RT, la bibliothèque ARTEMIS a été développée afin de permettre leur utilisation dans des simulations

temps réel en améliorant la précision et la vitesse de la simulation. Cela est fait à l'aide d'une méthode d'intégration appelée ART5. Plusieurs autres méthodes qui permettent la parallélisation et le découplage de certaines parties du calcul ont été également conçues dans le même but.

- RT-Events et RTE-Drive : Ensemble de blocs Matlab/Simulink de fonctions logiques et mathématiques dans le cas de RT-Events et de composantes d'électronique de puissance pour RTE-Drive. Leur principal avantage est le fait qu'ils comportent des méthodes permettant de compenser les erreurs qui se produisent entre les pas de temps et des méthodes améliorant l'interpolation des événements survenus entre deux pulsions d'horloge pour une meilleure précision lors des simulations temps réel ou les simulations hybrides.
- RT-Lab XSG : Les cartes d'entrées/sorties des stations de calcul comportent des cartes FPGA qui peuvent accueillir des parties bien spécifiques de la simulation ainsi réduire le pas de temps. RT-Lab XSG est un ensemble de blocs Matlab/Simulink qui permettent d'incorporer la génération code VHDL et la transmission du code C dans la simulation temps réel (afin de permettre la configuration des cartes FPGA).

1.4. Conclusion

Dans ce chapitre on a présenté les différentes techniques de simulation, on s'est intéressé à les clarifier théoriquement et d'expliquer les différences entre elles. L'emphasis a été portée sur la simulation temps réel hybride puisque c'est la méthode la plus efficace et qui se rapproche le plus du comportement réel des systèmes physiques spécialement en électrotechnique.

CHAPITRE 2

*PROPRIETES DE LA SIMULATION EN TEMPS REEL
AVEC OPAL-RTLAB*

2.1 Introduction

RT-LAB, produit phare d'OPAL-RT, est un logiciel de simulation temps réel qui conjugue performance et convivialité. RT-LAB a révolutionné le monde de l'ingénierie des systèmes, que ce soit dans le secteur spatial, terrestre ou naval. Il permet aux ingénieurs et scientifiques d'accélérer le développement de prototypes et de tester rigoureusement de nouvelles technologies et systèmes innovants. Entièrement intégré avec MATLAB/Simulink®, RT-LAB permet la conception de modèles élaborés en simulant leur interaction avec le monde réel. Par sa souplesse et son extensibilité, il prend en charge les applications de simulation temps réel les plus avancées dans les secteurs de l'automobile, l'aérospatiale, l'électronique de puissance et les réseaux électriques.

OPAL-RT Technologies une de ses caractéristiques essentielles est l'intégration parfaite avec MATLAB/Simulink. RT-LAB permet de convertir des modèles Simulink en temps-réel via Real-Time Workshop (RTW) et de les exécuter sur un ou plusieurs processeurs. Ce procédé est particulièrement utile pour la simulation hybride ou Hardware-In-the-Loop (HIL) et le prototypage rapide de systèmes de contrôle/commande. RT-LAB régit la synchronisation, les interactions avec l'usager, l'interfaçage avec le matériel à l'aide des cartes E/S ainsi que les échanges des données pour une exécution distribuée, de façon transparente pour l'utilisateur.

2.2 Simulateurs OPAL-RT [02]

Le simulateur d'OPAL-RT peut être divisé en deux parties. Une partie appelé Station hôte (*Host station*), qui contient le logiciel et qui permet de commander la simulation, d'interagir avec le système simulé et d'afficher les résultats. La deuxième partie est appelée station cible (*Target station*), qui se résume en un supercalculateur (Jusqu'à 64 coeurs) modulé avec des cartes E/S, numériques et analogiques (Les cartes comportent des modules de conversion analogique-numérique et numérique-analogique et des cartes FPGA). Cette deuxième partie contiendra le modèle à simuler et pourra être connecté à un prototype physique. Les stations cibles de calcul fonctionnent sous les systèmes d'exploitation temps réel QNX ou RedHat.

Une ou plusieurs stations de calcul (nommées aussi cibles), permettent d'exécuter les simulations en temps-réel. La configuration distribuée (plusieurs cibles) permet à des modèles complexes d'être distribués sur une grappe de PC fonctionnant en parallèle. Les cibles de la grappe communiquent ensemble avec des protocoles de communication tels que FireWire, SignalWire ou InfiniBand qui sont suffisamment rapides pour assurer une communication fiable en temps-réel. La grappe temps-réel est reliée aux stations de commande par un réseau TCP/IP comme indiqué sur la figure (2.1). Il est possible d'assembler et d'agrandir la grappe au besoin, puis de redéployer les PC pour d'autres applications lorsque la simulation est terminée. RT-LAB peut gérer jusqu'à 64 cibles s'exécutant en parallèle.

Dans les stations de calcul. Leur nombre et leur type dépendent des applications envisagées.

Il faut aussi mentionner l'existence des cartes actives, c'est-à-dire des cartes avec des processeurs FPGA (Field programmable gate array) intégrés, qui permettent par programmation directe l'obtention des modèles pouvant être exécutés à des pas de temps même inférieurs à la microseconde. L'interface avec l'utilisateur de RT-LAB est conçue de manière à parvenir à l'exécution en temps-réel en quelques clics.

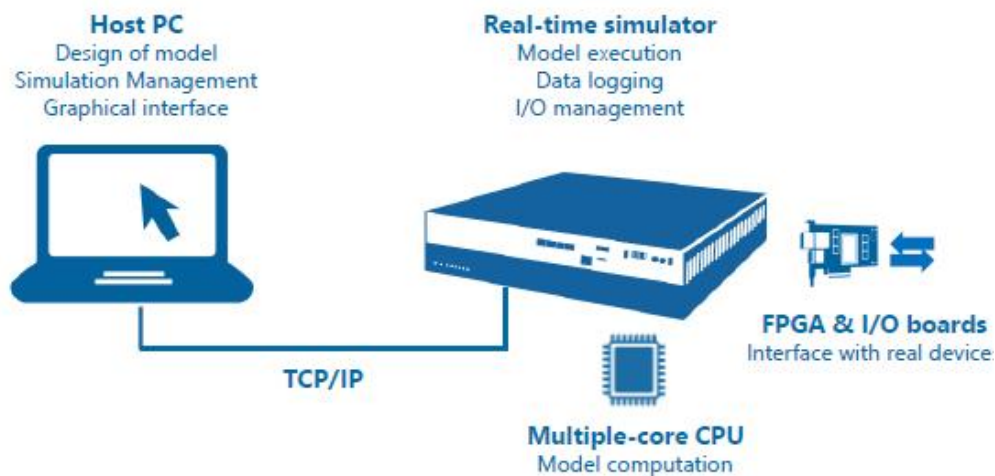


Figure (2.1) : Les deux hôtes (stations) de Simulations avec OPAL RT-LAB

2.3 Types de simulateurs développés par OPAL RT-LAB [02]

OPAL RT-LAB propose une série de simulateurs développés selon des améliorations et performances désirées : OP4510, OP4200, OP5030, OP5600, OP5607, OP5700, OP7020 ..ect. On s'intéresse dans ce document au simulateur OP4510 sur lequel on a réalisé nos modèles.

2.3.1 Le simulateur OPAL 4510

L'OP4510 est un simulateur compact d'entrée de gamme associant tous les atouts d'OPAL-RT en matière de prototypage rapide de commande (RCP) et de simulation avec matériel dans la boucle (HIL) haute performance.

Ce simulateur temps réel offre les meilleures performances à un prix abordable. Cette dernière déclinaison du modèle OP4500 marque une étape décisive en amenant à un autre niveau les capacités de connectivité, d'extensibilité et de polyvalence de la plateforme.

Plus que jamais, ce simulateur offre des meilleures performances et solutions aux applications en électrotechnique : réseaux électriques, l'électronique de puissance ...

Son architecture modulaire signifie qu'il peut être personnalisé via l'ajout d'une large gamme de cartes d'E/S (IOs). Son faible coût signifie qu'il est idéal pour les laboratoires de recherche ainsi que pour les besoins d'enseignement. « Pour les universités, ECADtools fournit des packages basés sur OP4510 avec un ou deux cœurs déverrouillés, ce qui en fait la solution idéale pour la modélisation de 25 à 50 nœuds de système d'alimentation. ».

2.3.2 Caractéristiques spécifiques

- Cible temps réel avec 2U puissantes avec processeur 4 INTEL CPU Cores à 3,3 GHz,
- Système d'exploitation en temps réel: Linux REDHAT,
- FPGA Xilinx KINTEX-7 avec I/O boards ;
- Total de 4 cartes I/Os : 16 A.in, 16 A.out, et 32 D.out et 32 D.in,
- Connecteurs DB37 arrière,
- 4 interfaces optiques SFP 5 Gbps en option.,
- Connecteurs de synchronisation,
- Emplacement de PC le pour la connexion avec les unités d'extension.

2.3.3 Architecture et périphériques

L'architecture (flexible) combinée commerciale FPGA et CPU standard de l'OP4510 utilise les dernières technologies permettant aux utilisateurs d'atteindre des pas de temps de simulation en temps réel inférieurs à **10 μ s et 20 ns**, respectivement.

La simulation entre FPGA et CPU est également possible, grâce à une liaison PCIexpress rapide échangeant des données et des signaux entre des périphériques. Cette fonctionnalité permet de coupler des modèles FPGA haute vitesse, tels que des convertisseurs de puissance et des entraînements électriques, à des systèmes électriques et mécaniques plus lents sur la CPU, offrant ainsi des simulations encore plus détaillées.

La simulation entre FPGA et CPU est également possible, grâce à une liaison PCIexpress rapide échangeant des données et des signaux entre des périphériques. Cette fonctionnalité permet de coupler des modèles FPGA haute vitesse, tels que des convertisseurs de puissance et des entraînements électriques, à des systèmes électriques et mécaniques plus lents sur la CPU, offrant ainsi des simulations encore plus détaillées.

La résolution temporelle de **10 nanosecondes** du FPGA permet aux utilisateurs de réaliser des RCP avancés de contrôleurs à haute vitesse basés sur PWM pouvant être testés sur du matériel réel grâce à un OP4510 complet avec un système de test de contrôleur d'électronique de puissance autonome et des modèles électroniques de puissance prédéfinis pour que les projets soient opérationnels en moins de temps que jamais auparavant.

La figure(2.2) illustre l'architecture du simulateur pour chaque option à l'aide de vues d'assemblage des composants matériels de simulation dans le châssis OP4510 contenant un support FPGA pouvant accepter quatre cartes mezzanine OPAL-RT standard, en plus aux signaux RS422.

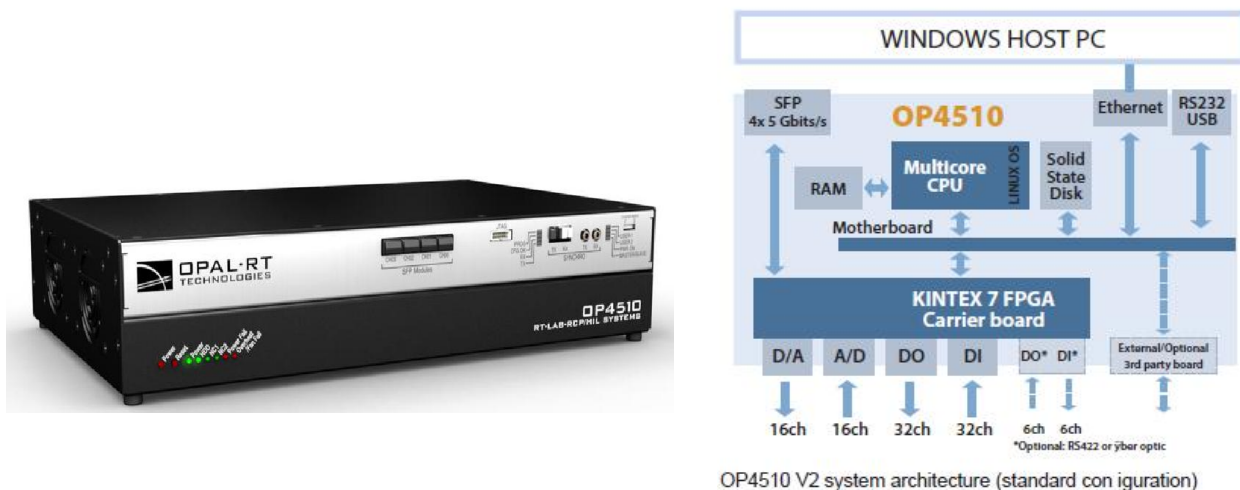


Figure (2.2) : Vue frontale du simulateur OP4510 + Architecture interne

La figure (2.3) montre la face arrière de l'OP4510 où sont connectés tous les périphériques externes :

A. Connecteurs DB37 pour entrées et sorties numériques ou analogiques.

B. Dans la configuration standard OP4510 standard, cette section contient une plaque vierge, cependant, plusieurs options sont disponibles. «Options d'extension d'E / S OP4510 V2».

C. Connecteurs d'ordinateur standard (de gauche à droite): souris et clavier, ports USB, moniteur, ports réseau (Ethernet).

D. Module connecteur DB9 pour signaux RS422, généralement utilisé pour transmettre les entrées de codeur différentiel et les sorties.

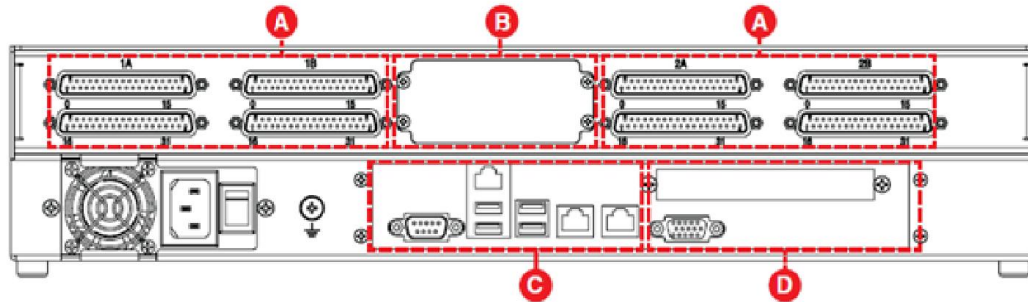


Figure (2.3) : Vue d'arrière de l'OP4510

Les caractéristiques des groupes entrées / sorties sont données au tableau (2.1).

Sorties numériques	Entrées numériques	Sorties Analogiques	Entrées Analogiques
▪ 32 canaux ▪ isolation optique ▪ Tension de sortie définie par l'utilisateur (5 à 30V) ▪ Courant maximal +/- 50 mA par canal	▪ 32 canaux ▪ isolation optique ▪ Tension d'entrée flexible (4 à 50 V) ▪ Courant d'entrée 3,6 mA.	▪ 16 canaux asymétriques. ▪ Résolution 16 bits ▪ Plage de tension +/-16 V ▪ Temps de conversion 1µs	▪ 16 canaux différentiels ▪ Résolution 16 bits ▪ Plage de tension +/-20 V ▪ Temps de conversion 0.5 ou 2.5 µs

Tableau (2. 1) : Configurations des groupes E/S (I/Os) de l'OP 4510

2.4 Les Avantages de simulation avec RT-LAB [02]

❖ Cout réduit

- Pas besoin d'un vrai système ou d'un prototype.
- Détecter les défauts tôt : le plus tôt sera le mieux.
- Dysfonctionnement minimes après l'installation.

❖ Réduire les délais

- Développer indépendamment le **HW** et le **SW** d'un contrôleur.
- Systèmes de test indépendamment dans le laboratoire.
- Réduire les activités de reprise avec une vérification progressive.

❖ **Réduire les risques**

- Étudier un système complexe en détail avec simulation.
- Meilleure couverture de test.
- Tester le système dans des conditions défectueuses de manière plus sûre.

En plus :

- Fournir plus de services de haute gamme.
- Système de plus en plus intelligent.
- Envoi d'un système plus sûr.
- Combinez plusieurs systèmes.
- Des délais plus courts.
- Ressources limitées.



La figure (2.4) montre la combinaison control système qu'offre une simulation temps réel avec RT-LAB

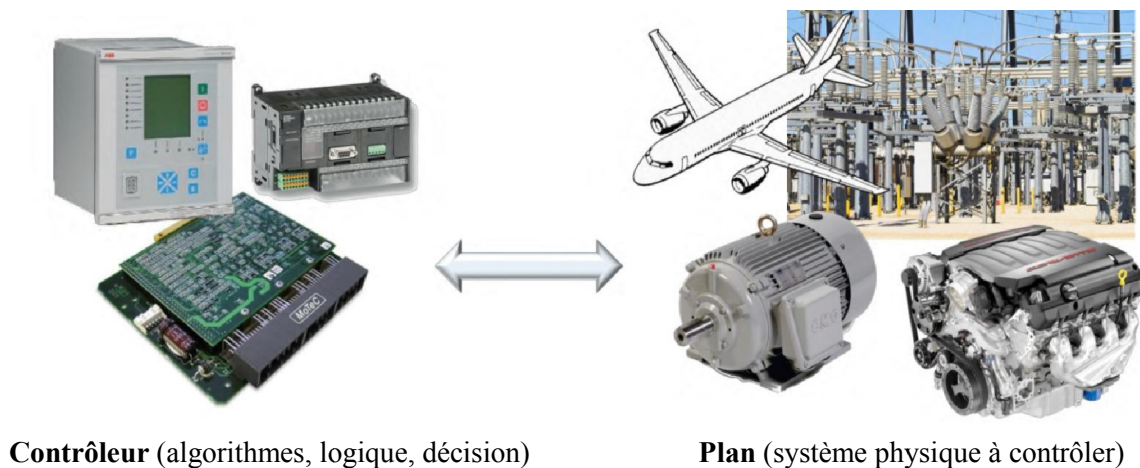


Figure (2.4) : Avantages d'application du simulateur RT-LAB

2.5 Concepts et apport de simulation en temps réel avec OPAL RT-LAB [02]

Simuler en temps réel c'est de : reconstituer virtuellement, à l'identique et en temps réel les conditions d'utilisation d'un système électrique ou mécanique à développer, étudier les conditions et causes de panne du système et les impacts à petite et grande échelle, détecter et corriger les erreurs de conception ou d'implémentation du système de contrôle avant la fabrication d'un prototype physique.

La simulation en temps réel sous RT-LAB veut dire :

- ✓ Donner le bon résultat,
- ✓ Au bon moment,
- ✓ Assez rapide uniquement en fonction de l'application.

Exemples dynamiques :

- Microseconde (10^{-6}) : Pour les transitoires électromagnétiques.
- Milliseconde (10^{-3}) : Pour les réponses temporelles mécaniques.
- Seconde : Pour le contrôle de la température (systèmes à des longues réponses)

Le temps réel est requis : figure (2.5-a)

- lorsque le matériel est connecté à la simulation.
- assure de réponses cohérentes dans un temps-déterministe.
- permet à la simulation d'assimiler le vrai matériel.

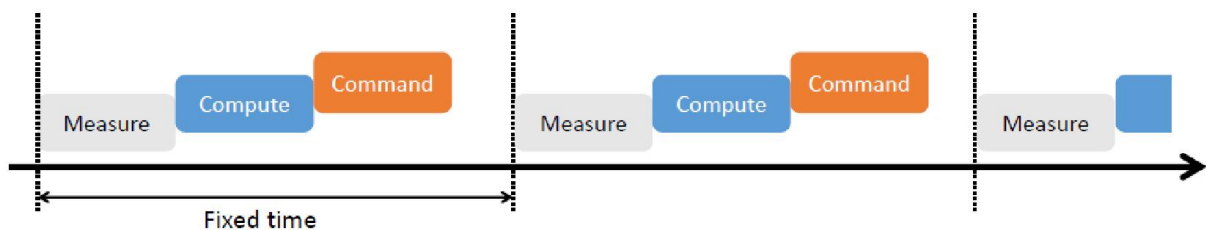


Figure (2. 5-a) : Concept de Simulation en temps réel

Le contrôleur doit prendre en compte les mesures, calculer l'algorithme de contrôle et générer des commandes dans un délai déterminé.

Et qu'est-ce qui se passerait s'il ya un dépassement ? Figure (2.5.-b)

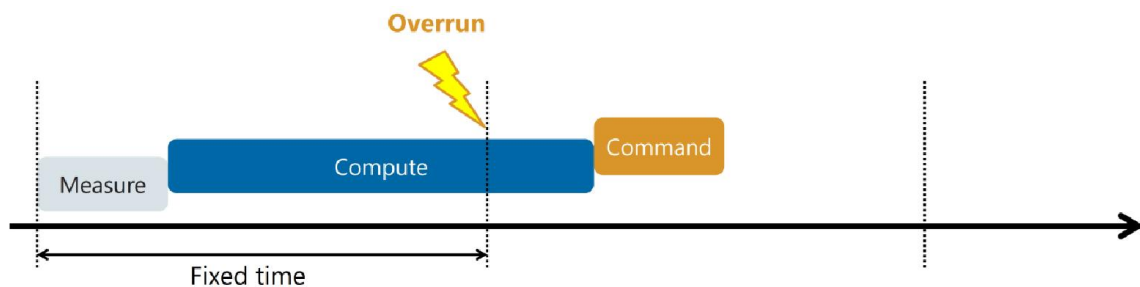


Figure (2. 5-b) : Concept de Simulation en temps réel

Trop tard... Le temps réel n'est pas respecté. Le système contrôlé peut ne pas se comporter comme prévu.

En conséquence, le temps réel:

- est nécessaire lorsqu'un matériel réel est connecté à la simulation
- assure des réponses temporelles cohérentes: déterminisme
- permet à la simulation d'attirer le vrai matériel

OPAL-RT fournit un simulateur numérique temps réel de gamme pour la simulation des modèles de différents domaines mécanique, hydraulique, électromécanique, énergétique, robotique en électrotechnique tel que les réseaux électriques, l'électronique de puissance et les convertisseurs, les entraînements électriques, les énergies renouvelables ...etc. Ces applications sont présentés sur les figures (2.6-a,b).

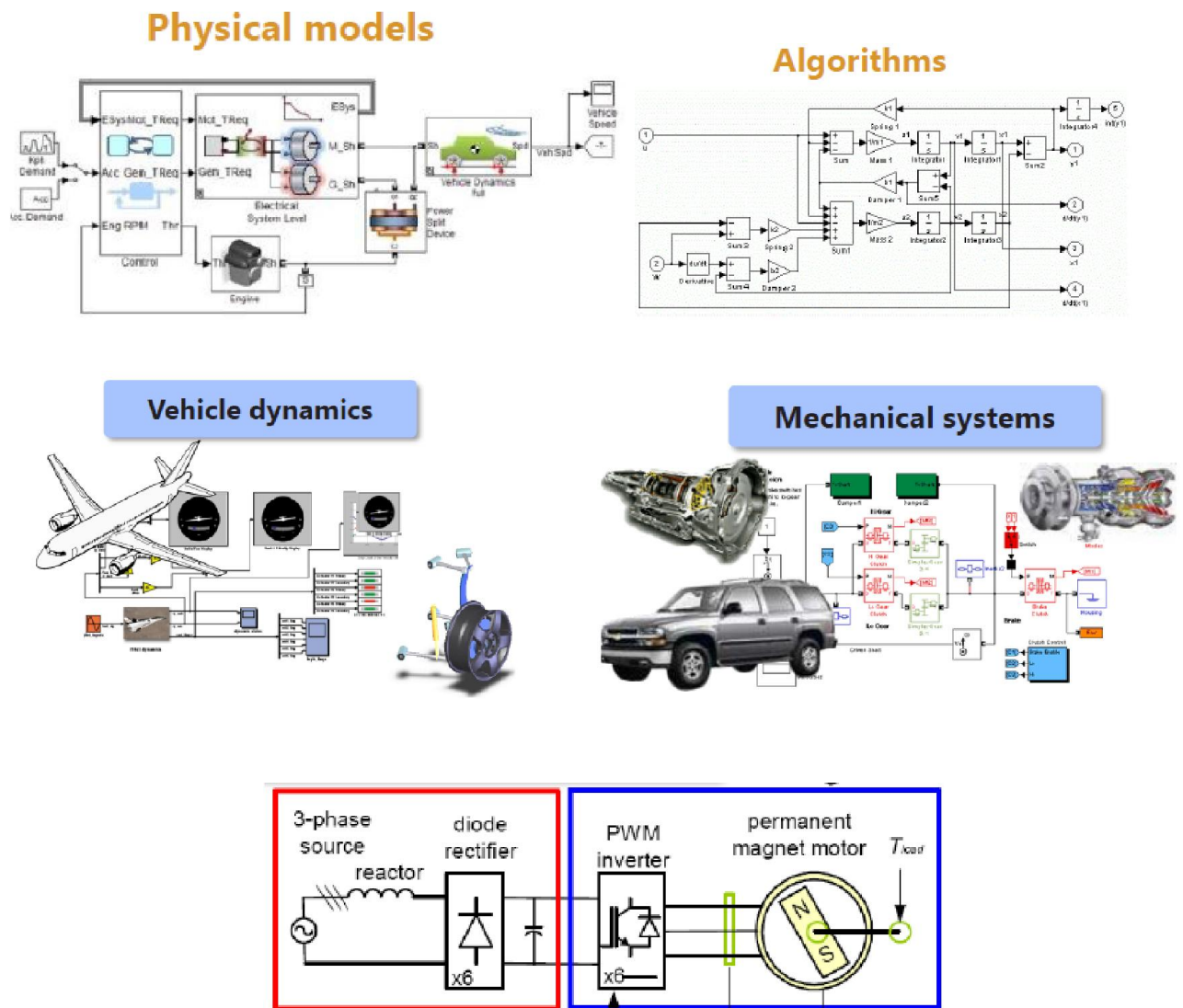


Figure (2. 6-a) : domaines d'application des Simulateurs OPAL RT-LAB

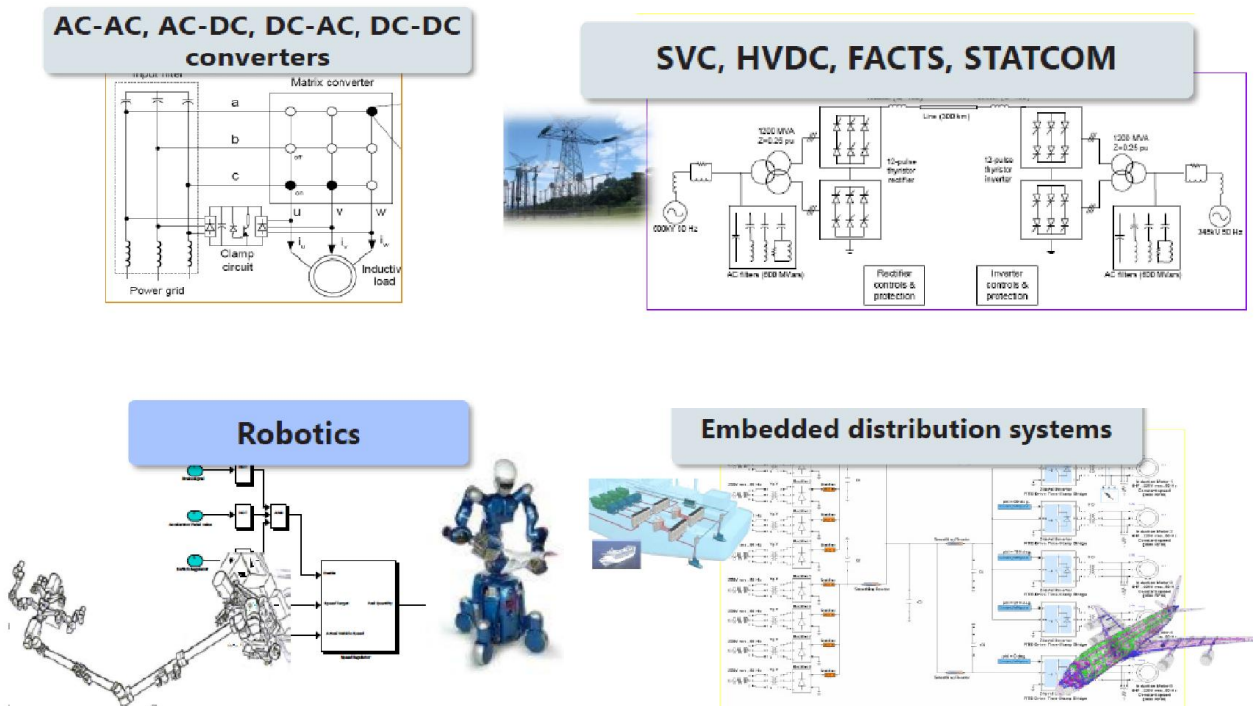


Figure (2. 6-b) : domaines d'application des Simulateurs OPAL RT-LAB

2.6 Les différentes méthodes de simulation en temps réel [02]

Les logiciels RT-LAB offrent une diversité de méthodes de simulation selon les exploitants et leurs objectifs, test de prototypes de systèmes complexe ou de contrôleurs, établissement de modèles fiable, d'où on distingue les structures suivantes présentées à la figure (2.7):

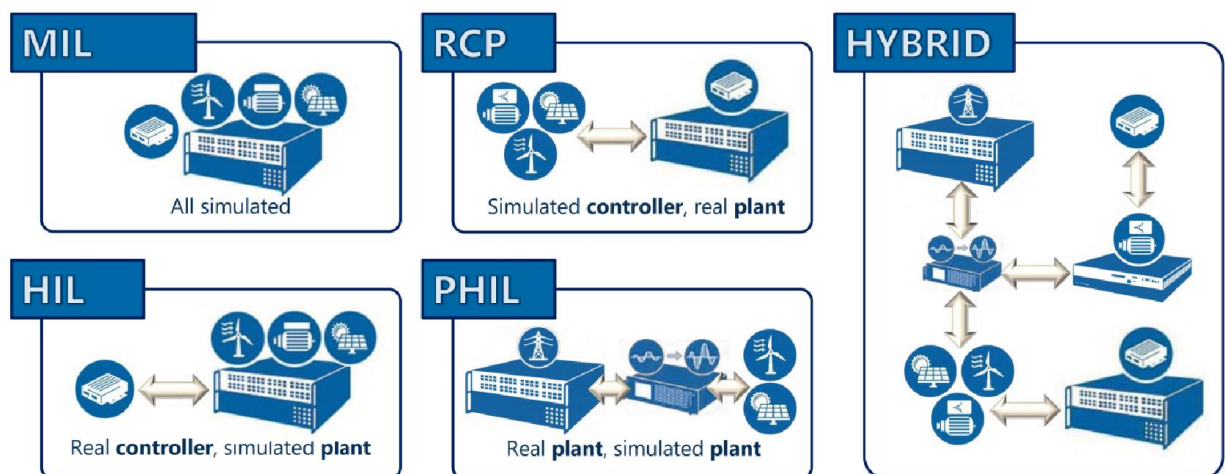


Figure (2. 7) : Méthodes de Simulation en temps réel avec OPAL RT-LAB

2.6.1 Modèle dans la boucle (Model In the Loop: MIL)

Dans cette technique tout est simulé, le système physique (Plant) et le contrôleur (Controller) sont tous simulés c.-à-d. représentés par des modèles. Voir la figure (2.8).

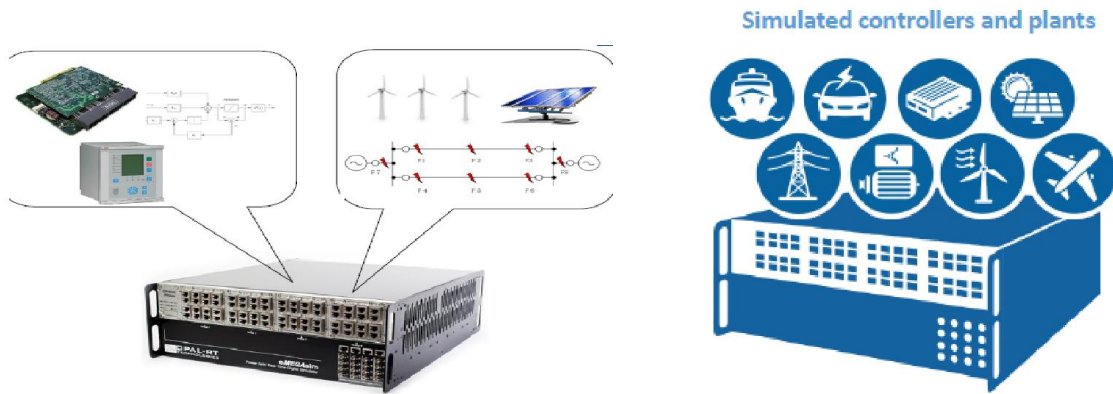


Figure (2. 8) : Modèle de simulation dans la boucle (Model In the Loop :MIL)

Objectifs :

- ✓ Lors de la conception d'un système
- ✓ études préliminaires.
- ✓ Accélération de la simulation.

Cas d'application

- Études de stabilité du réseau. Lorsqu'on a un réseau large, les valeurs sont en (RMS),
- Phaseurs avec une Stabilité électromécanique en *ms*.
- Régime transitoire d'origine d'électromagnétique sur le système de transmission d'énergie
- Réseau plus petit : Étude Harmonique – régime transitoire électromagnétique en μs

Cette méthode (MIL) nous permet d'obtenir des résultats plus rapides (Les heures de simulation deviennent des minutes), et plus courts (Simulation plus rapide = plus des itérations pour affiner un design). Et une détection précoce des pannes (Capturer rapidement les erreurs du système).

2.6.2 Prototypage de contrôle rapide (Rapid control prototyping : RCP)

Dans cette technique, le système physique (Plant) est réel et par contre le contrôleur est simulé (modèle). Les figures (2.9-a), (2.9-b) illustrent la structure de cette technique avec un exemple de développement.

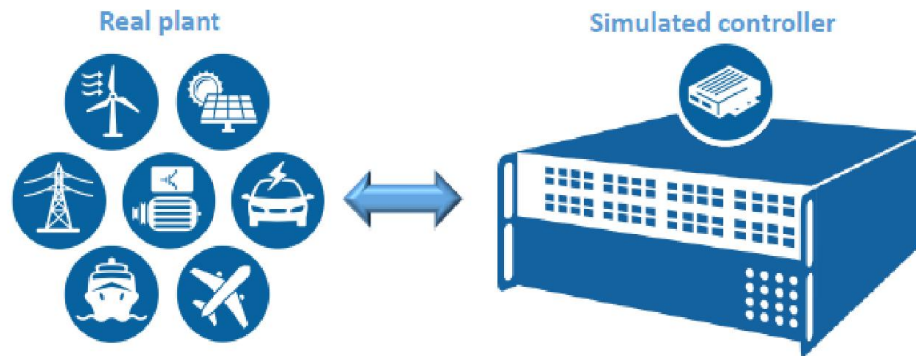


Figure (2. 9-a) : Technique de Prototypage de contrôle rapide (Rapid control prototyping : RCP)

Objectifs :

- ✓ Pour tester les algorithmes d'un contrôleur.
- ✓ Pour affiner les paramètres de l'algorithme.
- ✓ Pour connecter le contrôleur simulé à un plan réel.

Cas d'application : Développement de nouveaux algorithmes de contrôle sur des entraînements électriques existants.

Avec cette structure on peut :

- Découpler le HW et SW en développement d'un contrôleur : Le contrôleur hardware final n'est pas requis.
- Affiner l'algorithme de contrôle, Les paramètres de contrôle sont accessibles au moment de l'exécution.
- Détection précoce des pannes : Les erreurs de conception peuvent être capturées avant la mise en œuvre finale du contrôleur.

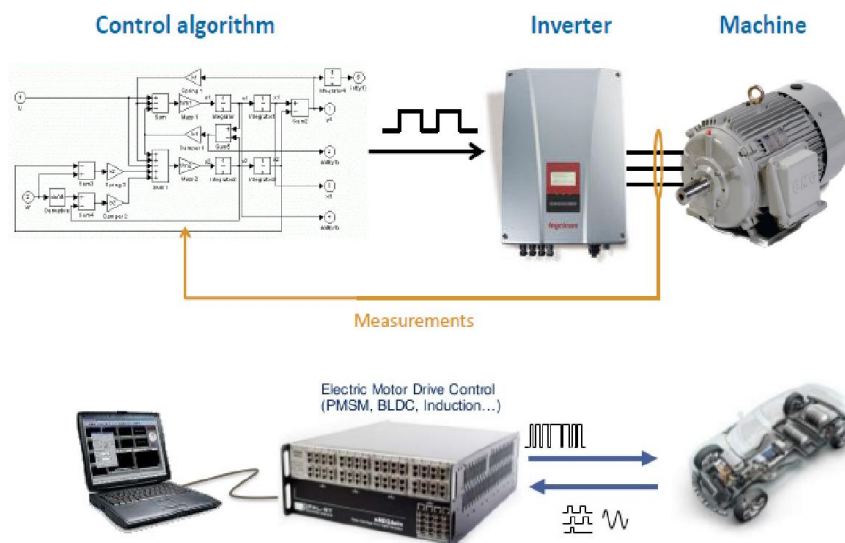


Figure (2. 9-b) : Exemple de Développement d'un algorithme de contrôle sur une machine existante

2.6.3 Matériel dans la boucle (Hardware In the Loop : HIL)

Dans cette technique, représentée à la figure (2.10), le système physique (Plant) est simulé (modèle) et par contre le contrôleur est réel.

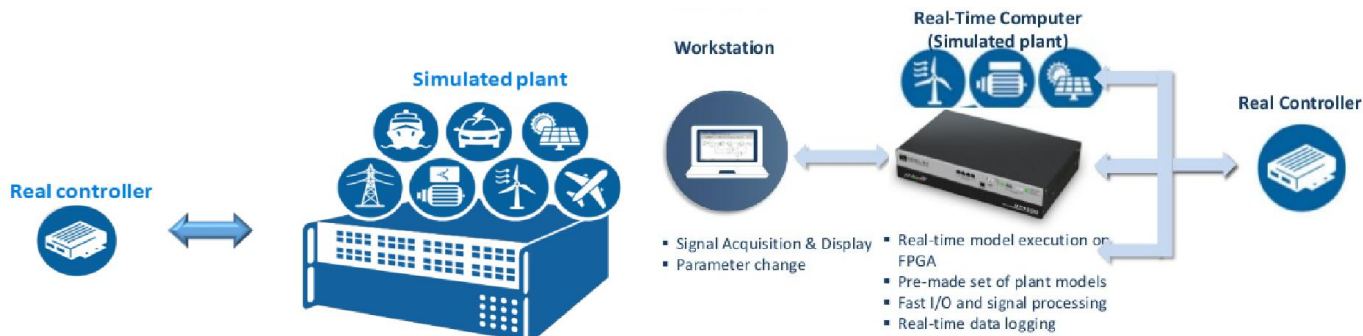


Figure (2. 10) : mode de simulation HIL: Matériel dans la boucle

Objectifs:

- ✓ Tester le contrôleur final dans les conditions sûres.
- ✓ Préparer les derniers tests physiques.
- ✓ Pas besoin que le vrai plan soit disponible.

Cas d'application

- Test d'un relais de protection (réel) dans un réseau électrique (modèle).
- Test de la commande d'un entraînement électrique.

La technique « HIL » est convenable pour :

- Valider l'intégration du HW et SW : les tests sont effectués sur le contrôleur final.
- Des tests plus sûrs (sains), le plan (système physique) est modélisé : Des tests difficiles ou dangereux peuvent être facilement effectués.
- Meilleure efficacité et couverture étendue des tests : des tests automatiques peuvent fonctionner 24/7 .Test à non-régression.

Sur la figure (2.11) sont illustrés quelques exemples du mode de simulation en temps réel selon cette technique (HIL).

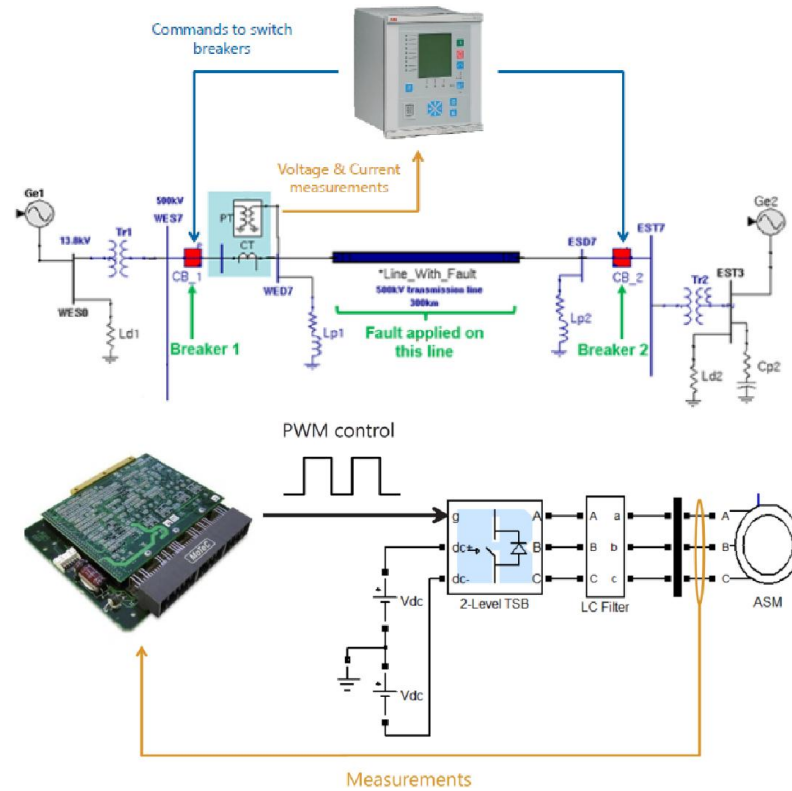


Figure (2.11) : Exemples de Simulation avec la méthode – HIL : Matériel dans la boucle

2.6.4 Matériel de puissance dans la boucle (Power Hardware In the Loop : PHIL)

Pour cette méthode, le système physique (Plant) est partiellement simulé et le contrôleur est peut être simulé ou bien réel comme le montre la figure (2.12).

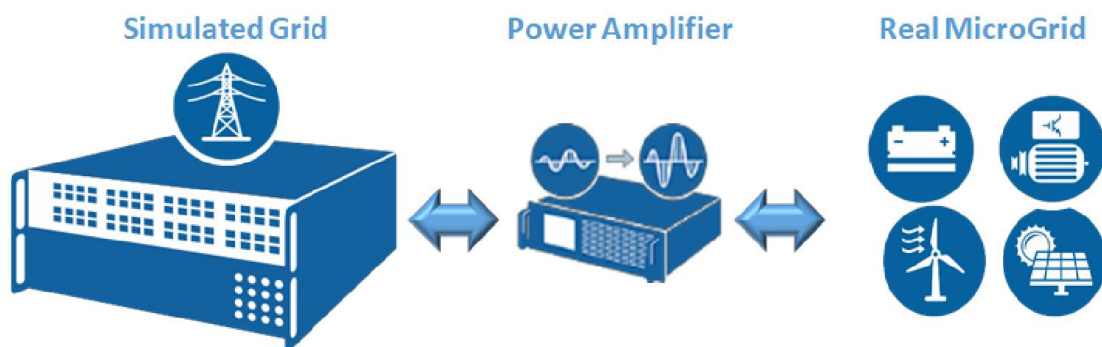


Figure (2.12) : mode de simulation PHIL: simulation : Matériel de puissance dans la boucle

Objectifs :

- ✓ Tester l'intégration entre les systèmes gérant de la puissance.
- ✓ Émuler les périphériques d'alimentation et leur environnement.

Cas d'application : La figure (2.13) représente l'intégration d'un micro réseau dans le réseau électrique.

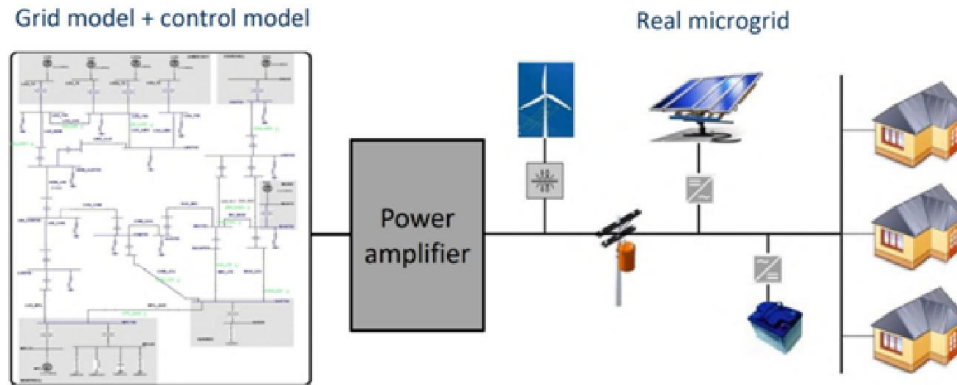


Figure (2. 13) : Exemples de Simulation avec la méthode PHIL

Ce type de simulation permet :

- D'émuler des dispositifs d'alimentation : Par exemple le réseau électrique.
- de Valider l'intégration d'équipement de puissance ou d'alimentation et de leur environnement : des tests sont plus proches de la réalité car une partie du système est réelle.
- De tester Les appareils qui doivent être connectés à des appareils d'alimentation ou de puissances : Test de PMUs et relais de protection.

2.6.5 Simulations hybrides complexes (Complex hybrid simulations :HYBRID)

Pour cette technique, le système (Plant) peut être simulé (modèle) ou réel (physique) et de même pour le contrôleur, il est simulé ou réel. La figure (2.14) représente ce mode de simulation.

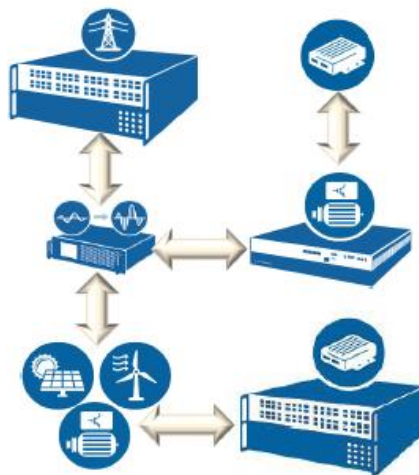


Figure (2. 14) : Simulations hybrides complexes : HYBRID

Objectifs :

- ✓ Tester des configurations complexes impliquant tout type de contrôleurs et de plan

Cas d'application : intégration d'un micro réseau dans le réseau électrique avec contrôleur réel et modélisé. Voir figure (2.15).

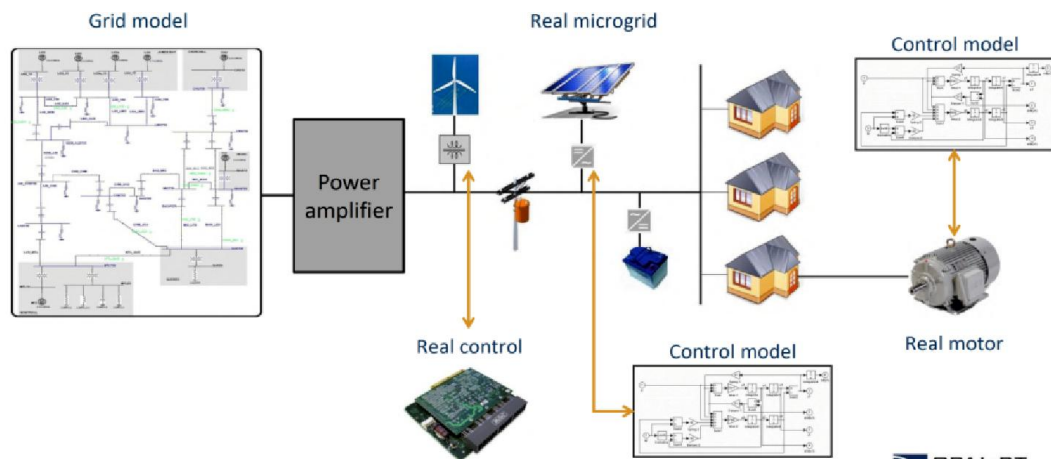


Figure (2. 15) : Exemple de Simulations hybrides complexes : HYBRID

2.7 Package logiciel et Outils de simulation avec les simulateurs OPAL-RT [02]

OPAL-RT propose une offre complète de simulation temps réel et de prototypage rapide pour une multitude de systèmes électriques, électroniques et mécatroniques : réseaux électriques, convertisseurs de puissance, commandes de moteurs électriques...etc.

Ces produits intègrent une interface logicielle et une plateforme matérielle à la pointe de l'innovation. Elles sont spécialement conçues pour mener des études de faisabilité, mettre au point de nouveaux concepts, concevoir et tester les contrôleurs.

Une multitude d'applications est possible, dont les convertisseurs de puissance, les entraînements électriques et hybrides, les très grands réseaux et les microréseaux intégrant des sources d'énergies renouvelables. La figure (2.16) représente le package logiciel d'OPAL-RT et les applications couvertes par chaque logiciel selon la période (fréquence) des phénomènes transitoires simulés qui peut varier de (1s) pour un grand réseau électrique jusqu'à (10ns) pour une simulation des convertisseurs statiques (électronique de puissance) sur FPGA.

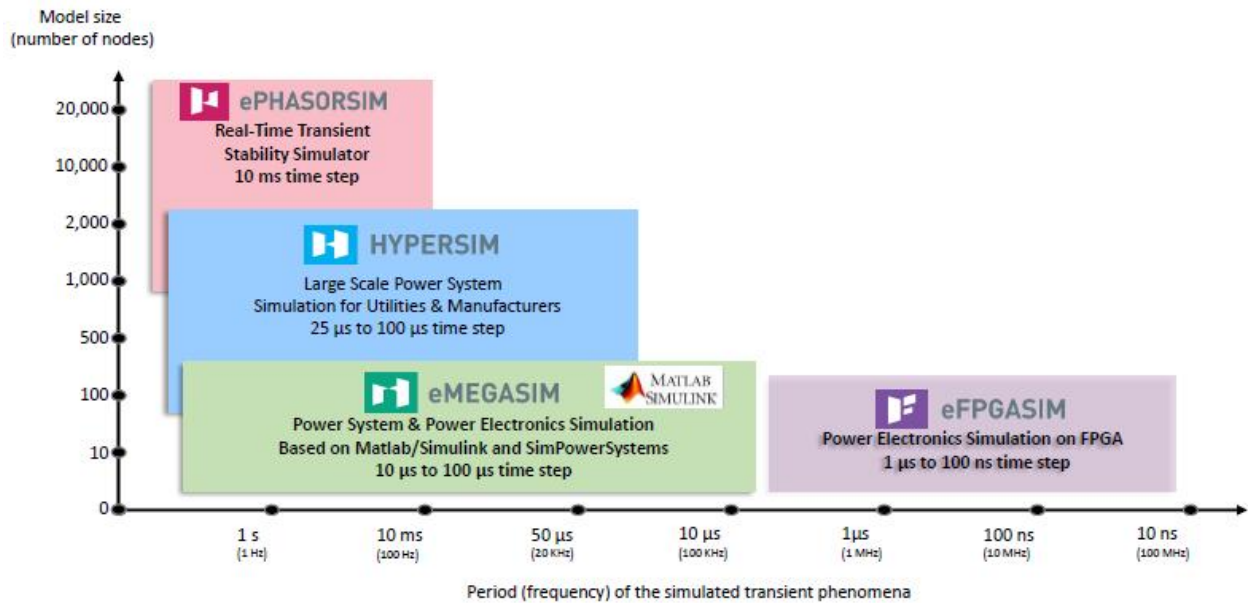


Figure (2.16) : Pack logiciel (SOFT) de simulation temps réel d'OPAL RT-LAB

La diversité du pack logiciel des systèmes de simulation temps réel est fondée sur la plateforme logicielle RT-LAB selon les applications tels que :

2.7.1 HYPERSIM : « *Le simulateur des réseaux électriques (de demain)* »

Le passage d'une production d'électricité centralisée à des réseaux de ressources énergétiques décentralisées (DER) constitue une révolution. Accompagné par l'intégration de technologies intelligentes offrant des capacités élaborées de surveillance, commande et protection en temps réel, ce changement de modèle augmente considérablement la complexité des réseaux denses de transport et de distribution d'électricité et des réseaux électriques localisés. Les nouvelles technologies promettent une fiabilité et une efficacité accrues mais représentent sans conteste de sérieux défis d'intégration.

HYPERSIM est l'outil idéal pour relever ces défis techniques. Sa plateforme perfectionnée de simulation temps réel permet des tests avec matériel dans la boucle (HIL). Que ce soit pour développer, intégrer ou tester des nouveaux composants et systèmes, ou tout simplement pour s'éviter des heures d'attente pour quelques secondes de simulation, HYPERSIM est la solution.

OPAL-RT développe en permanence HYPERSIM depuis 2010 et en a fait le simulateur de réseau électrique en temps réel le plus performant. OPAL-RT continue son étroite collaboration avec Hydro-Québec, ce qui permet à HYPERSIM de rester à la pointe de

l'innovation. Il est la référence au sein des laboratoires nationaux, des centres de recherche-développement industriels et des exploitants de réseau.

2.7.1.1 Les composants principaux de HYPERSIM

- **Logiciel HYPERSIM** : Élaborer des schémas unifilaires lisibles grâce à une bibliothèque complète de composants validés par Hydro-Québec et RTE.
- **ScopeView** : Créer des tableaux de bord, réalisez des calculs volumineux et complexes, publiez des rapports PDF détaillés.
- **TestView** : Automatise les tests et l'analyse pour une production rapide de rapports d'étude au format Excel ou PDF.

La figure suivante illustre un exemple de simulation en temps réel sous HYPERSIM

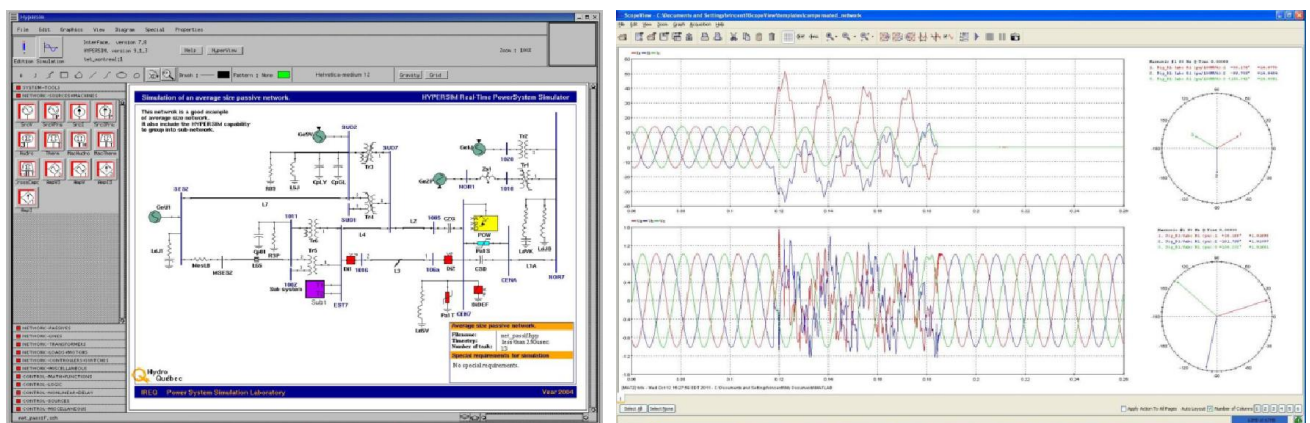


Figure (2.17): Simulation temps réel des réseaux électriques avec HYPERSIM

2.7.1.2 Applications typiques

- **Puissance** : HYPERSIM permet de simuler des réseaux électriques de plus de mille bus triphasés, en temps réel sur un seul simulateur, sans perte de précision.
- **Efficacité** : HYPERSIM offre le meilleur rapport coût / performance du marché des simulateurs temps réel.
- **Rapidité** : HYPERSIM permet de simuler des réseaux électriques sur des processeurs, et, dans le cas de composants d'électronique de puissance à haute vitesse comme des convertisseurs modulaires multiniveaux (MMC) sur des cartes FPGA.
- permet aux ingénieurs de créer rapidement des modèles de systèmes d'alimentation en utilisant à la fois des modèles de composants vérifiés dans le monde réel de la

bibliothèque HYPERSIM, ainsi que leurs propres modèles de réseau, d'appareil et même d'électronique.

- peut être exécuté sur l'ordinateur de simulation en temps réel, ou sur un ordinateur de bureau (sans HIL) en temps non réel pour le développement et la vérification de modèles
- s'intègre à eHS - le puissant outil de simulation électronique en temps réel d'OPAL-RT

2.7.2 eMEGASIM

Est un système flexible de simulation temps réel de réseaux électriques et d'électronique de puissance, pour les tests avec matériel dans la boucle (HIL ou Hardware-In-the-Loop) et le prototypage rapide de commande (RCP).

C'est une solution extensible, simple à utiliser et abordable, qui s'adapte à une grande variété de domaines et d'applications. Que ce soit pour un usage industriel ou universitaire, les utilisateurs peuvent ainsi élaborer le système de simulation temps réel qui répondra exactement à leurs besoins actuels et futurs.

Des réseaux de transmission et de distribution aux convertisseurs électriques et aux moteurs, eMEGASIM offre un environnement tout-en-un fondé sur l'outil Simulink pour la recherche-développement et les essais. Sa plateforme souple et personnalisable s'adapte à une large gamme de domaines et d'applications. Les équipements, logiciels et protocoles d'OPAL-RT sont extensibles et évoluent en continu pour nous permettre d'augmenter les performances de notre système et de diversifier nos applications dans le temps. La figure (2.18) représente la plate forme de simulation avec eMEGASIM.

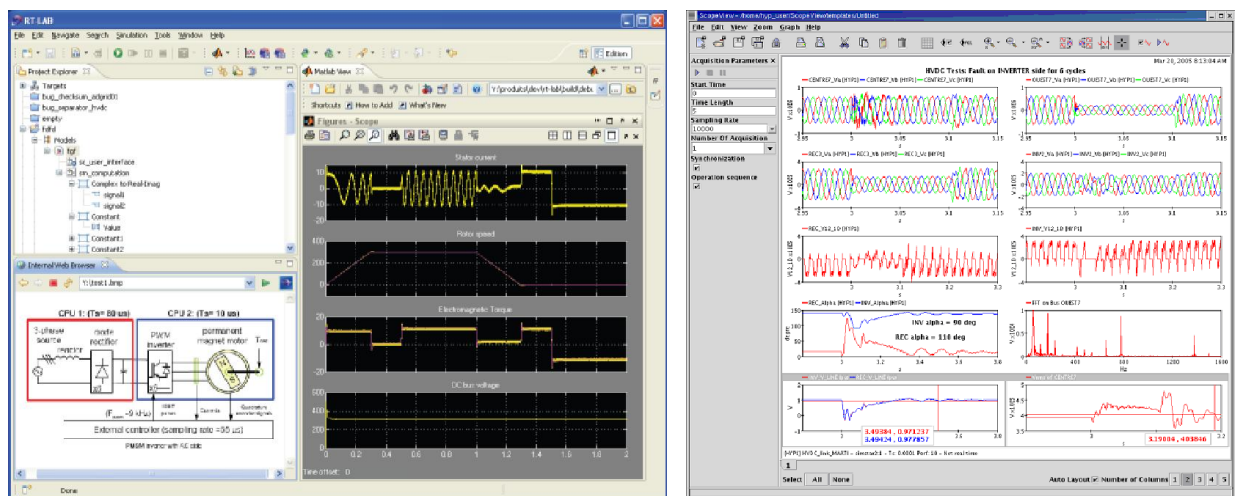


Figure (2.18): Simulation temps réel avec eMEGASIM

- **Les atouts d'eMEGASIM**

- **Prise en main rapide :** eMEGASIM fonctionne sur la plateforme de simulation temps réel RT-LAB, entièrement intégrée avec l'interface Simscape Power System, anciennement SimPowerSystems, facile d'utilisation et considérée comme une référence du secteur. Le temps de prise en main après acquisition en est ainsi indéniablement réduit. De plus, cette plateforme est aujourd'hui largement connue des étudiants.
- **Précision :** Extensible de 6 à 64 processeurs et bénéficiant de liaisons de communication à haute vitesse de 10 Gbits/s, eMEGASIM peut simuler des modèles avec des pas de temps de seulement 10 μ s tout en garantissant des résultats rapides et précis. Associé à des coprocesseurs FPGA et à la boîte à outils RT-XSG d'OPAL-RT, il peut réaliser des simulations avec un pas de temps de l'ordre de quelques nanosecondes.
- eMEGASIM dispose des solveurs les plus rapides et les plus précis: ARTEMiS et ARTEMiS-SSN. Leurs algorithmes éliminent les délais artificiels tout en appliquant des techniques de pointe de découplage pour assurer une vitesse et une efficacité accrues.

2.7.3 RT-LAB

- **RT-LAB** est une plateforme perfectionnée de commande de simulation qui comprend une bibliothèque complète de modèles pour toutes les applications et tous les secteurs. Elle permet de créer, compiler, charger, exécuter et analyser des modèles.
- **RT-EVENTS** est une boîte à outils Simulink qui améliore la vitesse et la précision des simulations discrètes de systèmes à événements. Il s'agit de la solution idéale simulation de signaux de modulation haute fréquence pour les commandes de convertisseurs. RT-EVENTS applique une correction d'erreur pour prendre en les événements qui se produisent entre deux instants d'échantillonnage, et permet de plus la simulation distribuée en temps réel.

Pour une polyvalence maximale, les plateformes **eMEGASIM** sont directement compatibles avec **ePHASORSIM** et **eFPGASIM**, pour couvrir aussi bien les simulations des réseaux électriques étendus que celles de systèmes d'électronique de puissance de haute précision. La figure suivante (2.19) présente les différents logiciels fonctionnant sous RT-LAB. (installés dans RT-LAB)

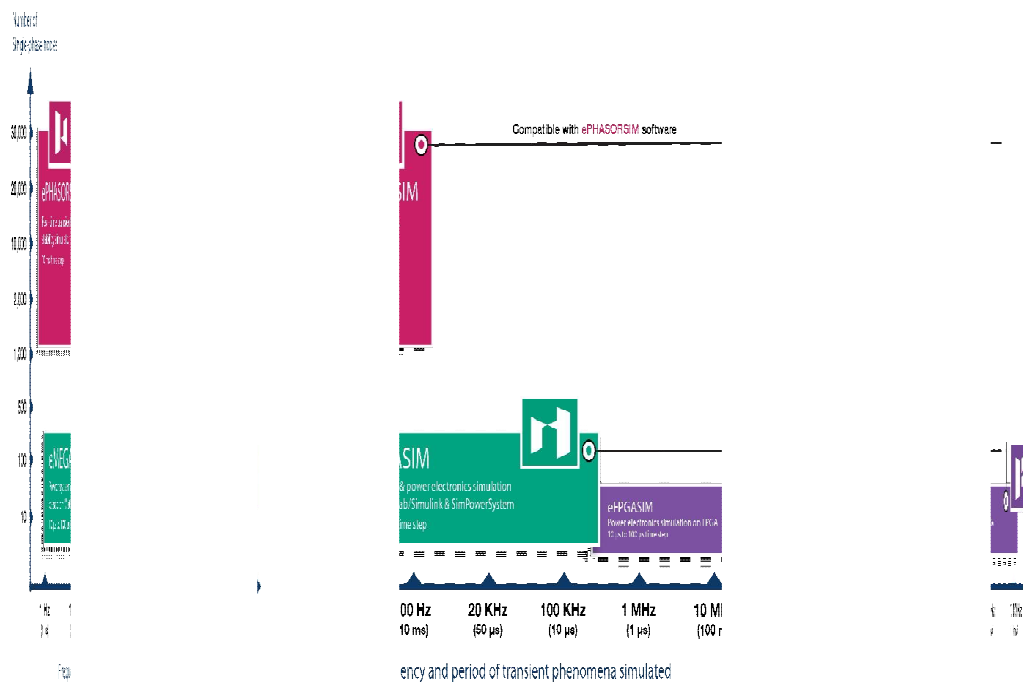


Figure (2.19) : fréquences des phénomènes transitoires simulés par le package contenu dans RT-LAB

2.7.3.1 Applications typiques

- Formation des opérateurs de réseau de transport à l'aide de modèles de réseau à séquence positive équilibrée
- Formation des opérateurs de réseaux de distribution à l'aide de modèles de réseau phase par phase non équilibrés
- Test et optimisation des commandes de la machine (par exemple, régulateurs de tension, régulateurs de vitesse, stabilisateurs de système d'alimentation) sur la stabilité du système
- Test de systèmes de contrôle mondiaux mis en œuvre sur SCADA à l'aide de mesures PMU pour le contrôle de la tension et de la puissance
- Test des systèmes de contrôle locaux tels que les changeurs de prises de transformateur, la commutation des condensateurs, la protection contre les surintensités et les fermetures, le délestage, le rétablissement de la charge et leur interaction avec les systèmes de contrôle mondiaux
- Test des interactions entre les systèmes de transmission FACTS et HVDC sur la stabilité du système des systèmes interconnectés à l'aide de la simulation de fréquence fondamentale avant de réaliser des simulations EMT détaillées
- Évaluations dynamiques de la sécurité du cybersystème des réseaux électriques à grande échelle
- Éducation et recherche.

2.7.4 eFPGASIM

La simulation temps réel de systèmes d'électronique de puissance reste l'un des plus ambitieux défis de la simulation avec matériel dans la boucle (HIL). Les capacités d'entrées/sorties pour la capture des signaux à largeur d'impulsion (MLI, ou *PWM*), la résolution mathématique de commutateurs couplés et l'injection de fautes à tous les niveaux d'un système complexe d'électronique de puissance, sont autant d'exemples illustrant la complexité de ce secteur en pleine évolution.

Après presque deux décennies de recherche-développement en simulation temps réel et d'expériences pratiques en électronique de puissance, OPAL-RT a lancé eFPGASIM, la solution temps réel sur cartes FPGA la plus puissante et la plus intuitive du marché. eFPGASIM est une plate-forme HIL de pointe qui associe la performance d'un simulateur numérique haute fidélité à une latence de communication très faible pour mettre au point et tester des systèmes de commande et de protection en électronique de puissance. La figure (2.20) représente un exemple d'application de la simulation avec eFPGASIM sous RT-LAB.

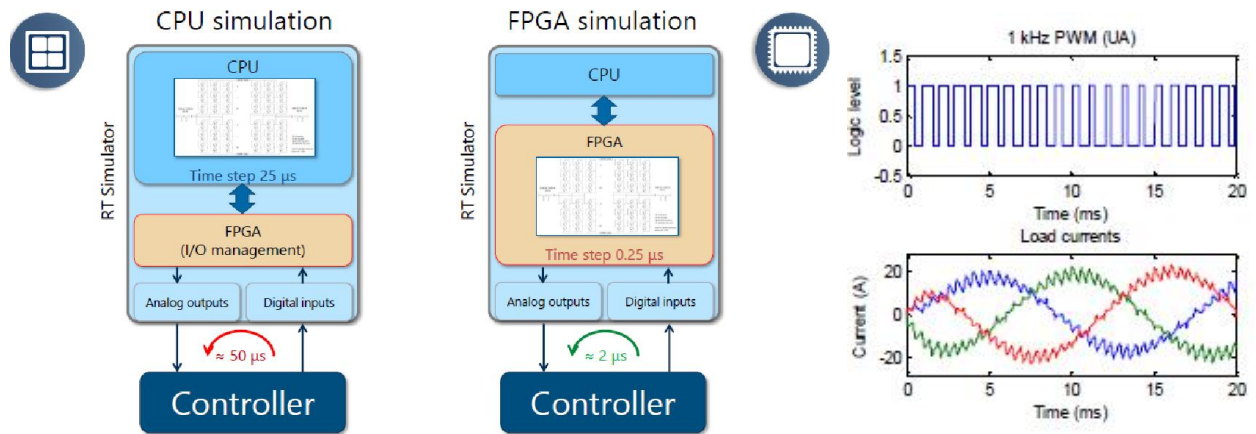


Figure (2.20): Simulation avec eFPGASIM sous RT-LAB

Le système eFPGASIM permet aux ingénieurs d'aller plus loin dans leurs tests grâce à son intégration avec d'autres produits, tels que:

- **eMEGASIM**, qui permet aux utilisateurs de connecter leur appareil électronique de puissance d'un microréseau à un très grand réseau électrique.
- **ePHASORSIM**, pour interagir avec et aborder l'intégration des énergies renouvelables et le réseau de transport progressivement surchargé, les problèmes de cybersécurité et la complexité de l'exploitation, de la surveillance, du contrôle et de la protection sur les réseaux électriques.

Protocoles de communication tels que Modbus, CEI 61850, DNP3, cAN, ARINC-429, MIL-STD-1553 et bien d'autres.

Une API complète et bien documentée, permettant aux utilisateurs d'interagir avec d'autres applications, telles que Python, TestStand, labVIEW, C++ et Java

2.7.4.1 eFPGASIM pour les transports électriques

Le simulateur numérique eFPGASIM constitue une plateforme HIL de pointe pour les tests de performance de systèmes de commande et de protection, avec une latence de communication très faible. Il permet aux ingénieurs en électronique de puissance de mettre au point des systèmes d'entraînement électrique avec un pas de temps inférieur à la microseconde.

2.7.4.2 eFPGASIM : boîte à outils logicielle

▪ *Moteurs électriques*

Une simulation temps réel pour des tests HIL ou PHIL (simulation avec matériel de puissance dans la boucle) de commande de moteur électrique doit être la plus fidèle possible à la réalité, non seulement pour maximiser la couverture des tests mais aussi pour privilégier la simulation temps réel afin de réduire les coûts engendrés par l'usage d'un dynamomètre.

À cette fin, eFPGASIM offre une vaste bibliothèque de modèles de moteurs électriques : machine synchrone à aimant permanent (PMSM), machine à induction (IM), générateur asynchrone à double alimentation ou DFIG, machine asynchrone à double alimentation ou DFIM, rotor à cage d'écureuil, moteur à réluctance commutée (SRM), moteur sans balais à courant continu (BLDC) et moteur à courant continu. OPAL-RT dispose aussi de l'expertise pour développer des modèles temps réel de moteurs atypiques, à la demande.

▪ *eHS (Electrical High Solver)*

eHS est un solveur électrique pour FPGA générique et reprogrammable qui s'inscrit au cœur de la solution eFPGASIM. Son interface utilisateur conviviale permet l'exécution en temps réel de modèles créés dans l'un des outils de simulation suivants : SimScape Power Systems, PSIM, PLECS ou Multisim. Avec la puissance d'eHS, eFPGASIM est le système idéal de simulation temps réel pour tout type d'applications de tests de conversion électrique, comme la conversion des énergies renouvelables (photovoltaïque, éolien, stockage de l'électricité et microréseaux), les systèmes d'entraînement industriels, les véhicules électriques et la recherche en électronique de puissance. Le principe de ce solveur pour FPGA est montré par la figure (2.21)

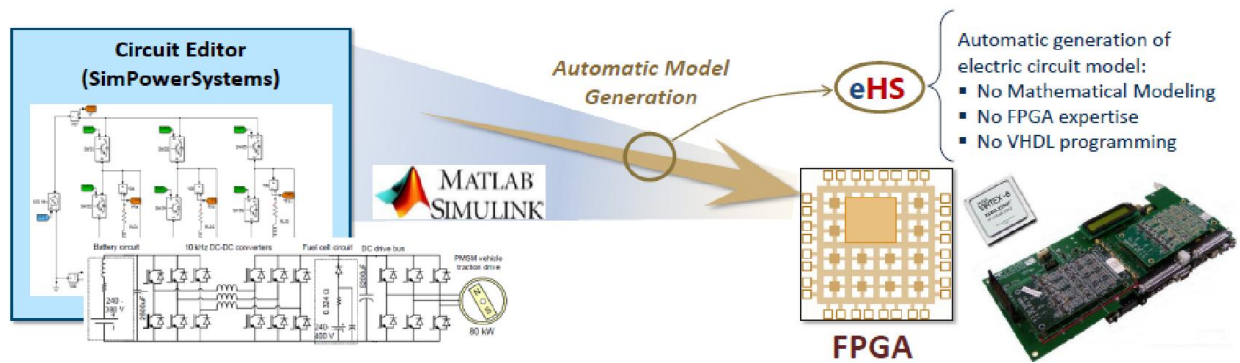


Figure (2.21) : eHS : Solveur électrique pour FPGA générique et reprogrammable

▪ Modèles de convertisseurs modulaires multiniveaux (MMC) sur FPGA

Le bloc MMC FPGA développé par OPAL-RT, simule les procédés de conversion modulaire multiniveaux de diverses topologies de sous-modules avec une très grande précision et une efficacité imbattable. Le modèle peut être intégré sans difficulté dans un plus large circuit électrique simulé dans **eMEGASIM** ou **HYPERSIM**. Tout comme d'autres blocs d'**eFPGASIM**, les utilisateurs accèdent directement au bloc **MMC FPGA** à partir de MATLAB/Simulink. Ils peuvent alors modifier dans les détails leur modèle et produire leur propre microprogramme FPGA grâce à la fonctionnalité RT-XSG qui permet aussi d'intégrer des interfaces particulières d'entrées/sorties et de personnaliser l'algorithme de contrôle dans la boucle. Le concept de modèles de convertisseurs MMC sur FPGA est illustré par la figure (2.22), la figure (2.23) montre la structure interne de la Co-simulation CPU/FPGA d'un modèle.

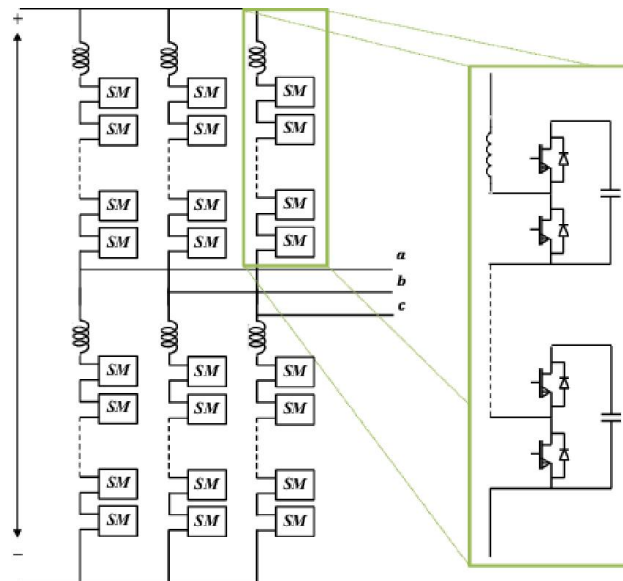


Figure (2.22) : Modèles de convertisseurs modulaires multiniveaux (MMC) sur FPGA

▪ RT-XSG

RT-XSG offre des blocs prêts à l'emploi, utilisables dans Simulink, pour la simulation FPGA avec matériel dans la boucle et le prototypage rapide de lois de commande. Il assure la configuration de la carte FPGA, de même que le transfert des données à haut débit entre les modèles de simulation RT-LAB et le système défini par l'utilisateur sur la carte. RT-XSG comprend une bibliothèque de plusieurs blocs aux fonctions comparables à celles des blocs MATLAB/Simulink® afin de faciliter la programmation sur cartes FPGA. Il permet également à l'utilisateur de générer un modèle personnalisé et de répartir le calcul sur plusieurs nœuds pour une simulation plus puissante et flexible.

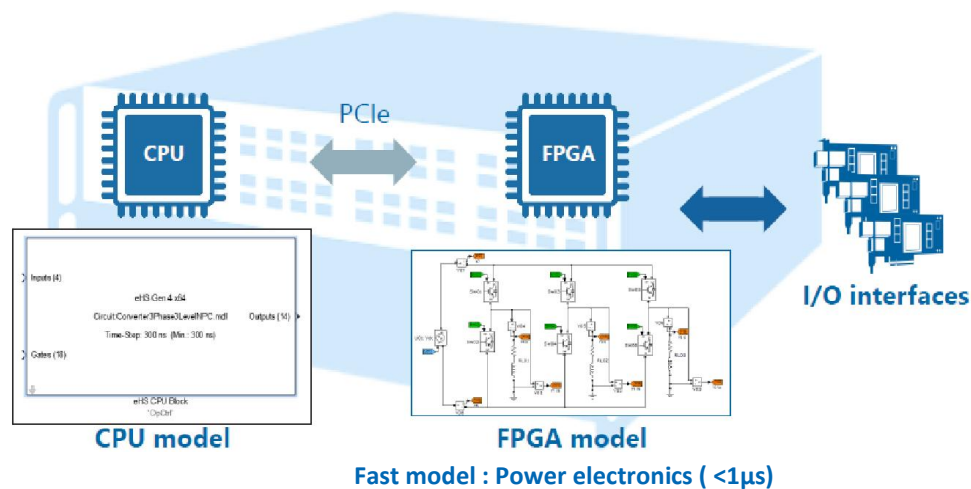


Figure (2.23) : structure interne : Co-simulation CPU/FPGA

2.7.4.3 Applications typiques

- Systèmes d'alimentation embarqués
- Transport hybride et électrique
- Commandes de conversion d'énergie

2.7.5 ePHASORSIM

« Simulation en temps réel de la stabilité électromécanique des grands réseaux électriques »

Le développement massif des sources d'énergie renouvelable, la saturation progressive du réseau de transmission et les questions de cybersécurité sont autant de facteurs qui augmentent la complexité de l'exploitation, la surveillance, la commande et la protection des réseaux électriques. La conduite de réseaux étendus, plus complexe, exige davantage de précision dans les évaluations et les études, ainsi que dans la formation à l'exploitation et à la validation des réseaux à venir. Des systèmes innovants comme ePHASORSIM permettent de

mener à bien ces opérations par une analyse approfondie des réseaux électriques, avec un gain de temps substantiel.

Fondée sur RT-LAB, environnement logiciel de référence d'OPAL-RT, la simulation temps réel ePHASORSIM permet la connexion à un modèle Simulink très complet afin que les chercheurs puissent bénéficier d'un écosystème de réseau exhaustif. Ils accèdent ainsi à tous les modules matériels et protocoles d'entrées/sorties et de communication d'OPAL-RT comme DNP3, OPC-UA ou C37.118, pour une interconnexion de la simulation temps réel avec les dispositifs physiques testés. La plateforme propose aussi toutes les fonctionnalités de surveillance, de traitement et d'acquisition des données pour faciliter les tests, les analyses et la création de rapports.

2.8 Conclusion

Dans notre travail on s'intéresse à la simulation temps réel avec les technologies d'OPAL RT-LAB, De ce fait, en cette partie on a présenté intérêt de l'emploi du simulateur RT-LB pour la simulation en temps réel et la diversité de ses applications dans les différents domaines technologiques spécialement en électrotechnique.

Une présentation de l'architecture interne et externe du simulateur OP4510 (HardWare) nous a permis de connaître les différents constituants du simulateur leurs fonctions ce qui détermine les performances de chaque simulateur.

Ensuite une description des différentes méthodes et types de simulation et le package logiciel (software) OPAL-RT-LAB était nécessaire afin de mieux comprendre les fonctionnalités qu'offre ces simulateurs.

La mise en œuvre de ces fonctionnalités, la création et la préparation d'un projet (modèle), sur l'interface d'utilisateur de RT-LAB conçue de manière à parvenir à l'exécution en temps-réel seront traités au chapitre qui suit.

CHAPITRE 3

ÉTAPES DE SIMULATION EN TEMPS REEL
SOUS RT-LAB

3.1 Introduction

Le développement logiciel est réalisé avec MATLAB/Simulink et RT-LAB, la plateforme de simulation temps réel développé par OPAL-RT. En général, le programme qui est exécuté dans le simulateur provient directement d'un modèle Simulink qui a été compilé avec RT-LAB. Voici les principales étapes de création d'un modèle exécutable pour le simulateur, toutes gérées avec RT-LAB (voir figure 3.1) :

1. Implantation et développement du modèle sous MATLAB/Simulink ;
2. Séparation du diagramme Simulink (modèle) en sous-systèmes pour l'assignation des processeurs de calcul ;
3. Génération du code C à partir du diagramme Simulink et du code MATLAB ;
4. Envoi du code C au simulateur ;
5. Compilation du code C et création de l'exécutable par le simulateur ;
6. Exécution du programme.

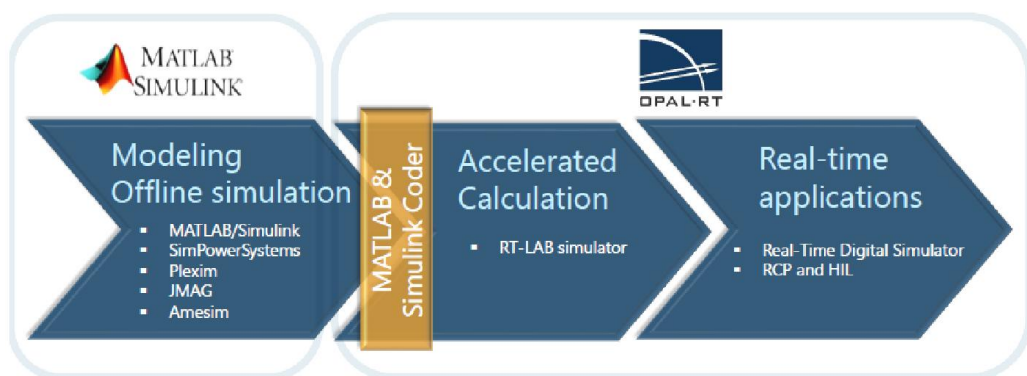


Figure (3.1) : Étapes de simulation sous RT-LAB

L'implantation Simulink (d'un modèle quelconque) qu'on présentera par la suite dans cette section peut être envoyée presque telle quelle vers le simulateur pour une exécution temps réel. Seules quelques modifications au diagramme Simulink original sont à effectuer à l'aide d'une

librairie Simulink qu'OPAL-RT met à la disposition des usagers. Parmi les modifications nécessaires, les plus importantes sont :

3.2 Structure du modèle [02]

3.2.1 Regroupement en sous-systèmes

Le modèle original (simulink) doit être séparé en sous-systèmes dépendamment de l'emplacement physique où chacun d'eux sera exécuté (assignation des coeurs de calcul). Trois types de sous-système peuvent être déclarés selon les deux premières lettres du nom du sous système (SM, SS ou SC). Le tableau (3.1) donne plus de détails à ce sujet.

Nomenclature	Fonction
SC_ anyName	pas de calcul, s'exécute sur le P.C. Sous-système de l'interface console (GUI subsystem) ; exécution asynchrone, donc à fréquence moins élevée que les sous-systèmes de calcul ; sert à l'échange de données avec la station de base (PC Windows). contient principalement : des scopes, des affichages, des constantes, des commutateurs manuels.
SM_ anyName	Sous-système de calcul principal (main subsystem) ; exécution synchrone au pas de calcul choisi. calcul en temps réel ou accéléré (mathématique, logic, génération des signaux I/Os, models physiques). Utilise un seul cœur du CPU du simulateur
SS_ anyName	Sous-systèmes de calcul additionnels ; exécution synchrone au pas de calcul choisi ; permet l'exécution d'une autre partie du modèle en parallèle au sous-système principal sur un autre cœur ; nombre maximal limité par le nombre de coeurs de calcul du simulateur.

Tableau (3.1) : Types de sous-systèmes RT-LAB/Simulink

Comme il est représenté sur les figures (3.2), (3.3), le modèle doit contenir que des blocs regroupés en sous systèmes « Ensemble de blocs qui sont placés dans un seul bloc appelé «sous-système» », ce qui permet de :

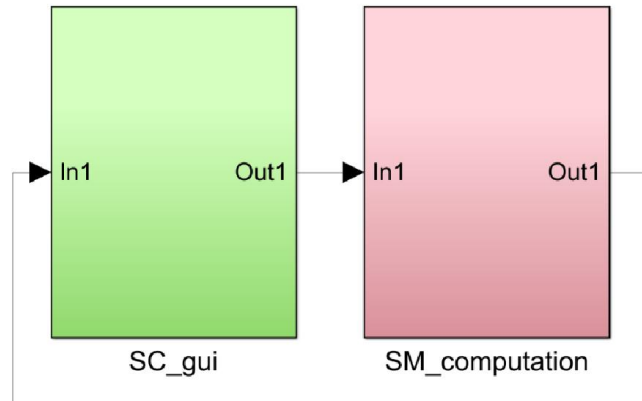


Figure (3.2) : Les deux principaux sous systèmes que doit contenir un modèle

- Simplifier le modèle en regroupant des blocs
- Établir un diagramme hiérarchique
- Gardez les blocs fonctionnellement liés ensemble

Dans les plates-formes RT-LAB, les sous-systèmes ont deux objectifs:

- a. Distinguer les sous-systèmes de calcul et le sous-système d'interface graphique
- b. Attribuer des sous-systèmes de calcul aux différents cœurs du processeur (CPU).

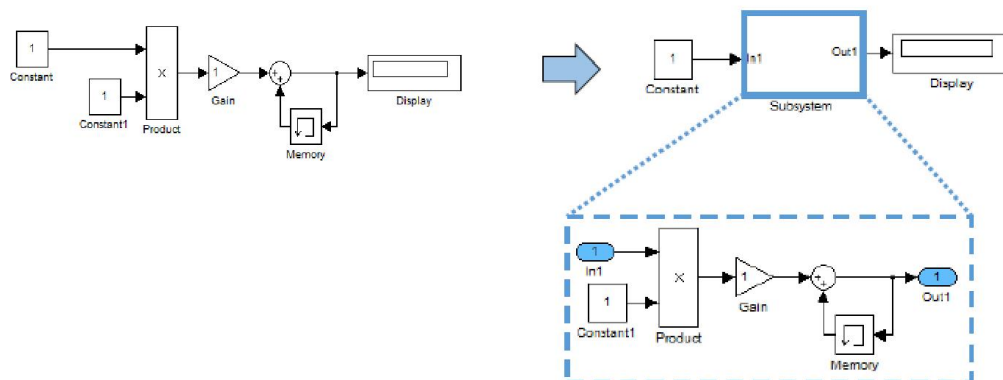


Figure (3.3) : regroupement des blocs en sous système

Quoi qu'il en soit, le niveau supérieur d'un modèle Simulink® utilisé avec RT-LAB doit afficher uniquement les sous-systèmes.

Les figures (3.4), (3.5) montrent la bonne méthode de regroupement des blocs de chaque sous-système.

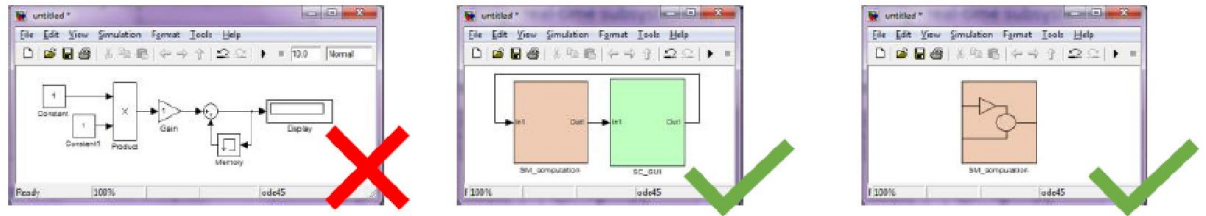


Figure (3. 4) : Distinction de l'interface de calcul et l'interface graphique.

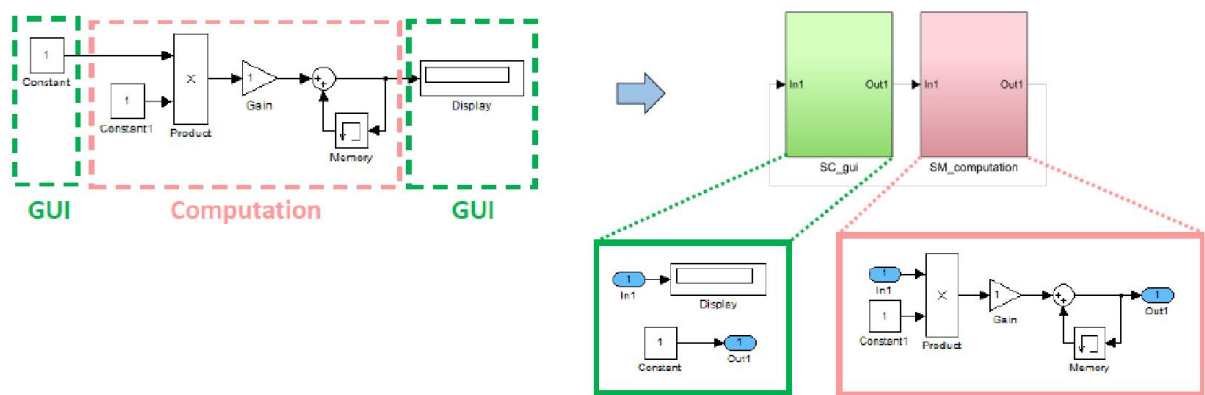


Figure (3. 5) : Regroupement des blocs et contenu de chaque sous-système

L'échange de nombreux signaux entre les sous-systèmes peut aggraver la lisibilité et la maintenabilité du modèle.

Des bus ou des signaux mixtes peuvent être échangés entre les sous-systèmes. il est préférable d'échanger un tel signal pour simplifier la vue de dessus du modèle et améliorer la maintenabilité.

3.2.2 Ajout de blocs OPCOMM

Toutes les entrées de chaque sous système défini doivent passer par ces blocs. Ils permettent la communication entre les cœurs de calcul. Pour chaque sous-système, un bloc opcomm doit être placé pour faire transiter les signaux de type synchrone et un autre pour les signaux asynchrones.

Un OpComm est responsable de la communication TCP/IP entre l'hôte et la cible. La figure (3.6) illustre le principe.

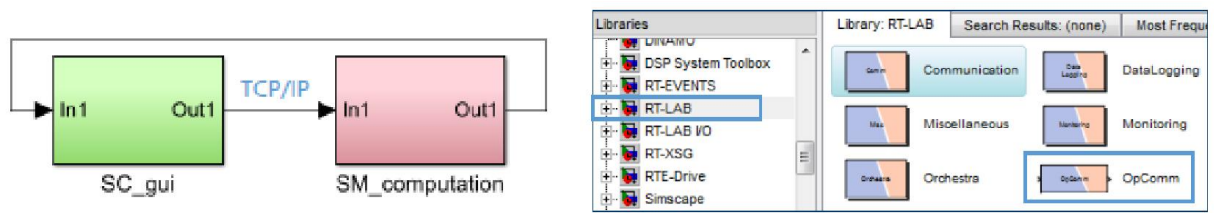


Figure (3.6) : Ajout de l'OpComm de la bibliothèque RT-LAB sous Matlab/Simulink

Afin que le modèle sera chargé et traité sur la plateforme RT-LAB, Il est indispensable d'insérer ou ajouter un 'OpComm' bloc responsable de la communication à partir de la bibliothèque RT-LAB qui sera disponible dans le navigateur de la bibliothèque Simulink®, une fois RT-LAB a été installé. Il est placé entre 2 sous- systèmes de calcul, ou bien entre sous-systèmes de calcul et sous-système d'interface graphique.

Alors un OpComm doit être placé à l'intérieur de chaque Sous-système SM, SC, SS, comme l'indique la figure (3.7).

Toutes les entrées de sous-systèmes (SM, SS, SC) doivent d'abord passer par un bloc OpComm avant toute opération sur les signaux auxquels elles sont associées. Le bloc OpComm doit être inséré après la création et le changement de nom des sous- systèmes (SM, SS, SC).

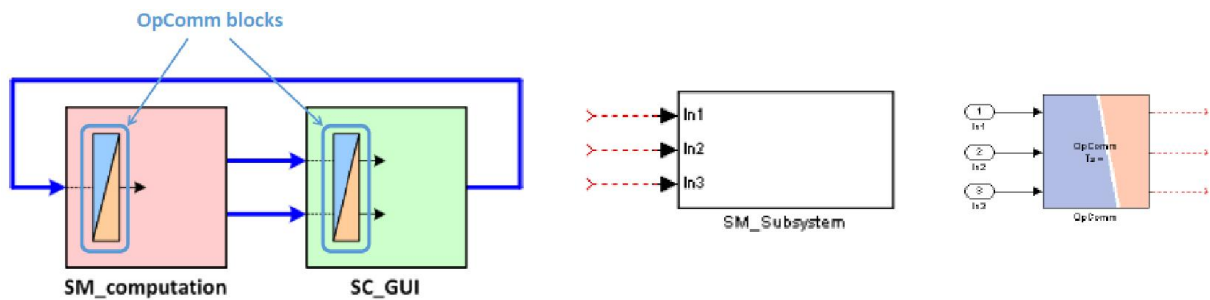


Figure (3.7) : OpComm inséré dans les deux sous systèmes

Un bloc OpComm peut accepter plusieurs entrées dans un sous-système. Double- clique sur le bloc pour sélectionner le nombre d'entrées requises. Chaque signal d'entrée peut être un scalaire ou un vecteur.

La figure (3.8) les différentes liaisons entre différents sous systèmes à travers l'OpComm.

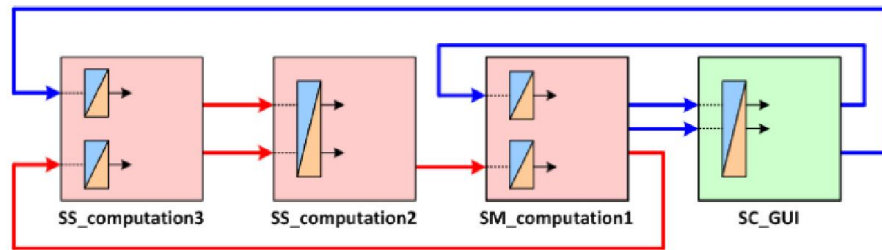


Figure (3.8) : communication entre sous systèmes via l'OpComm (rouge=synchrone, bleu=asynchrone)

Dans les sous-systèmes de calcul (SM ou SS)

- Un OpComm reçoit des signaux synchronisés en temps réel d'autres sous-systèmes temps réel.
- Un OpComm reçoit des signaux asynchrones du sous-système graphique

Dans le sous-système de console (SC)

- Un OpComm suffit dans la plupart des cas
- Plus de blocs OpComm (jusqu'à 25) peuvent être insérés pour recevoir les signaux du système temps réel sous-systèmes. Plusieurs blocs OpComm définissent des «groupes d'acquisition : jusqu'à 25» uniques avec leurs propres données paramètres d'acquisition (décimation, taille...)

La figure (3.9) montre l'assignation des sous systèmes de calcul aux cœurs (CPU) du simulateur, le sous système de calcul est attribué bien sur au Host PC.

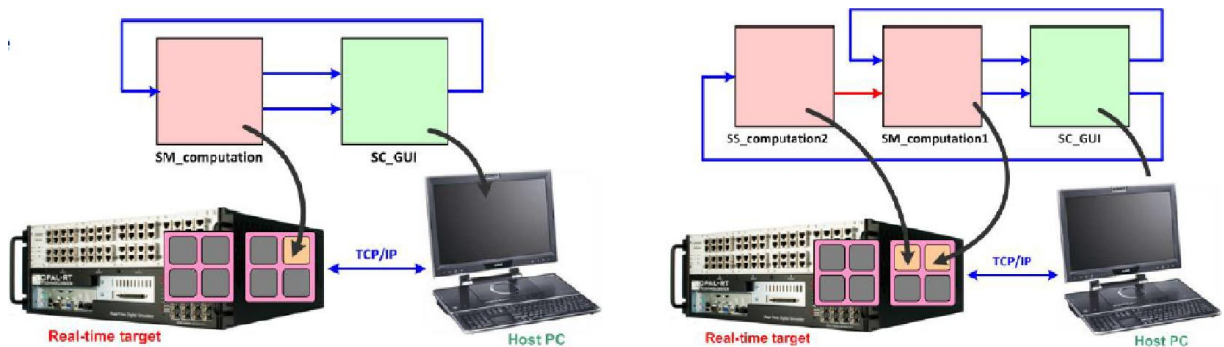


Figure (3. 9) : traitement des OpComm et occupation des cœurs du CPU du simulateur et Host PC

3.2.3 Ajout des blocs d'entrées- sorties

Plus de détails sont donnés à ce sujet plus loin dans cette section.

Par ailleurs, un ordinateur avec Windows est nécessaire pour utiliser RT-LAB, qui sert de station de base pour le simulateur. Une communication réseau entre l'ordinateur et le simulateur doit être établie. Depuis la station de base, il est possible d'interagir avec le

modèle s'exécutant en temps réel. Voici quelques exemples :

- Modifier l'état d'exécution du programme (arrêt, pause, marche, etc.) ;
- Visualiser les différents signaux du modèle ;
- Modifier des paramètres du modèle

Toujours à partir du modèle Simulink, RT-LAB offre des blocs Simulink qui permettent d'interagir avec des signaux physiques réels. Ces blocs agissent comme une interface avec les cartes d'entrées-sorties du simulateur temps réel.

Les entrées-sorties du simulateur sont gérées par l'intermédiaire d'un FPGA, comme montré au chapitre précédent. Celui-ci doit être programmé par un fichier bitstream qui fait la liaison entre les ports d'entrées-sorties physiques et le bus PC le vers le processeur principal.

Le fichier bitstream à utiliser dépend du modèle du simulateur utilisé. C'est un fichier (.bin), inclus avec l'installation de RT-LAB. De plus, un autre fichier du même nom (mais de type .conf) renseigne sur la configuration des canaux d'entrées-sorties réalisée par le bitstream. On peut l'ouvrir avec un éditeur de texte conventionnel. Ce fichier est important car il est analysé systématiquement par les blocs Simulink d'entrées-sorties.

Un sous-système de calcul principal (SM_SimulManager) peut avoir plusieurs fonctions :

- Acquisition des mesures provenant des capteurs ;
- Traitement des mesures en vue d'être envoyées vers le modèle.
- Écriture des signaux sélectionnés vers un fichier de sortie
- Affichage des signaux sélectionnés sur les sorties analogiques (I/Os).

3.3 Présentation du logiciel RT-LAB et la préparation d'un modèle [02]

En première phase, après lancement du logiciel RT-LAB, comme illustré à la figure (3.10), on doit Créer un nouveau projet :

- De l'icône,
- Dans le menu: Fichier> Nouveau> Projet RT-LAB,
- Depuis la vue Explorateur de projet.

On donne un nom de projet, puis on Clique sur "Suivant" pour ajouter un exemple de modèle à RT-LAB projet. On Clique sur "Terminer" pour créer un projet vide et ajouter manuellement un modèle.

Sur la figure (3.11) sont montrées les étapes de l'ajout d'un modèle exemple au projet sous RT-LAB.

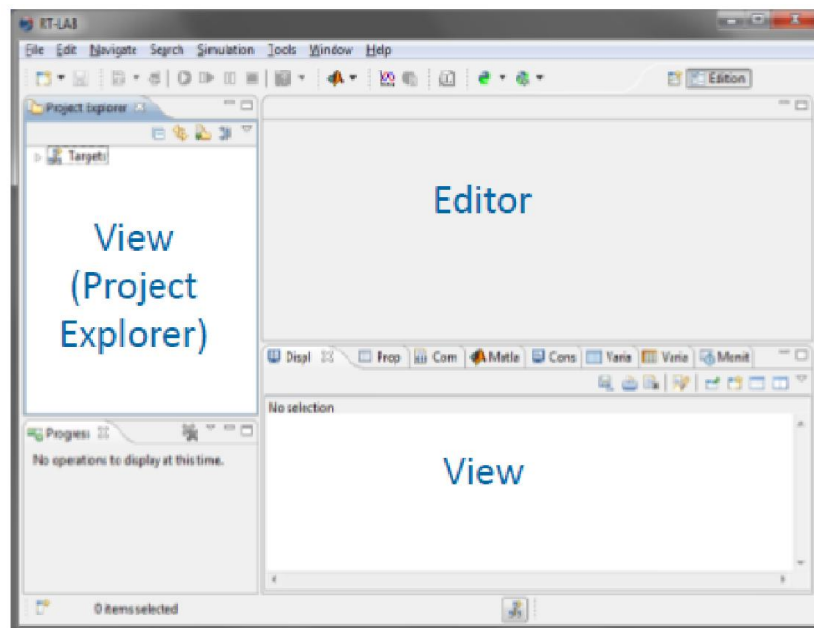


Figure (3.10) : fenêtre principale après lancement du logiciel RT-LAB

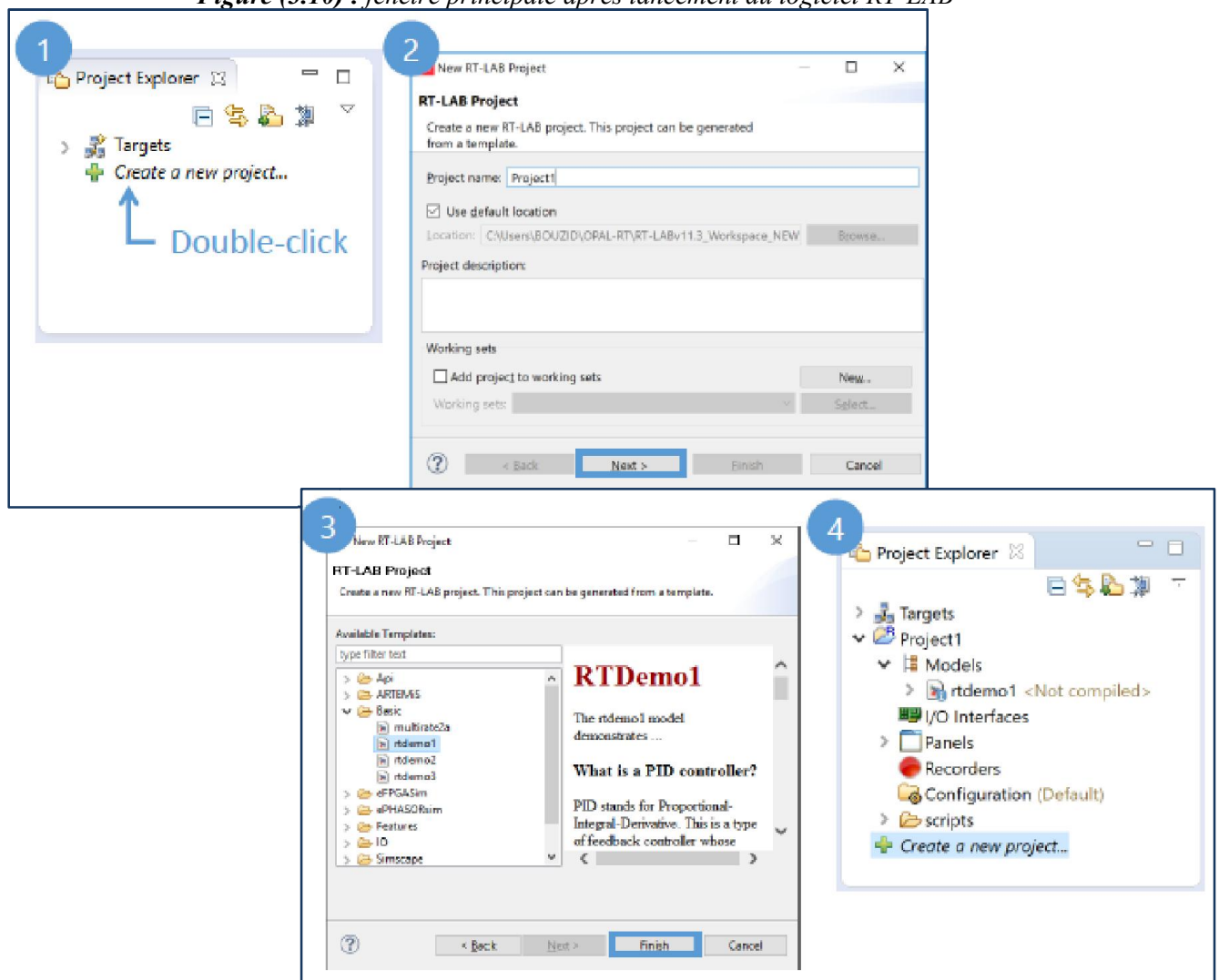


Figure (3.11) : Étapes de l'ajout d'un modèle exemple au projet sous RT-LAB

Le menu de RT-LAB nous permet aussi d'ajouter un exemple de modèle existant dans le projet. Comme exemple C:\OPAL-RT\RT-LABx.x.x\Exemples, illustré par la figure (3.12)

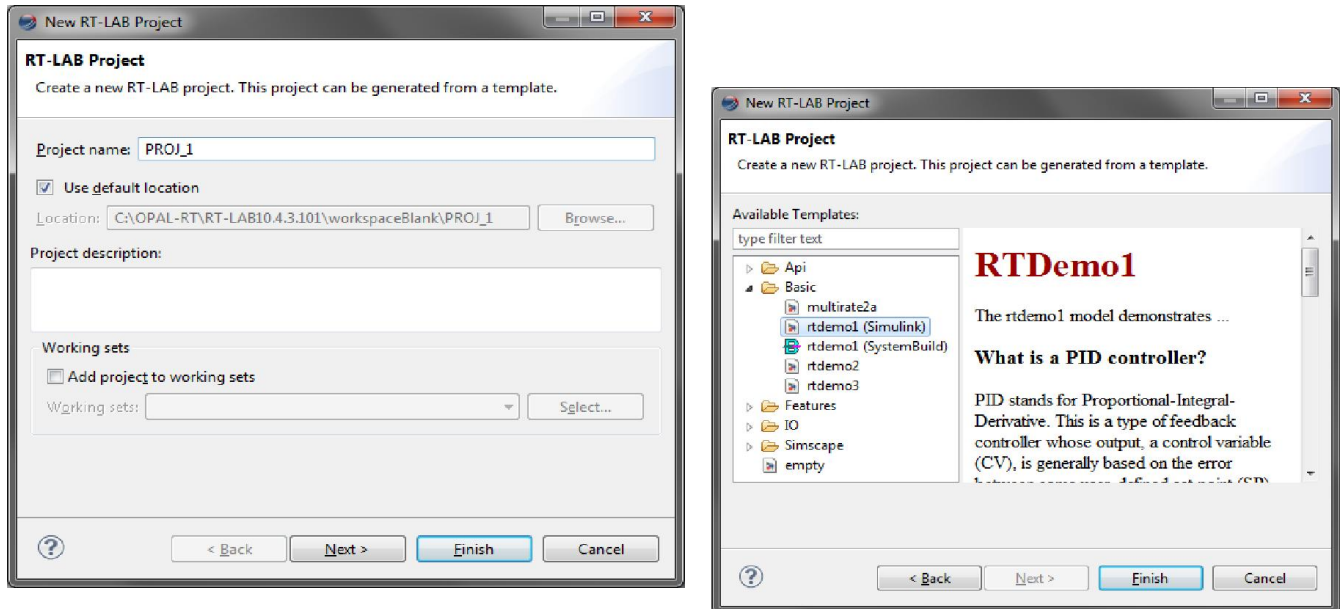


Figure (3.12) : fenêtres de l'ajout d'un modèle au projet sous RT-LAB

Comme indiqué sur la figure (3.13), on peut aussi ajouter au projet un nouveau modèle vierge ou sous-système par :

Fichier> Nouveau> Modèle RT-LAB

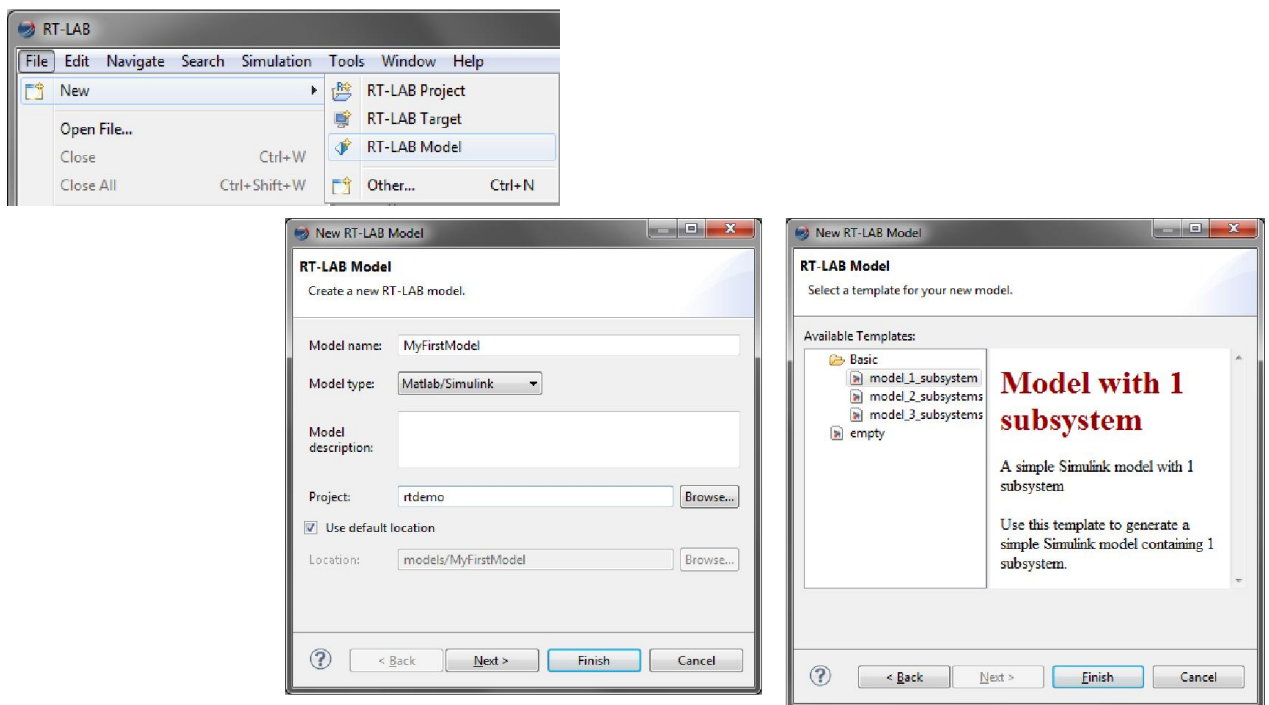


Figure (3.13) : Création d'un nouveau modèle vierge au projet sous RT-LAB

Comme il est possible d'ajouter un modèle d'utilisateur existant au projet :

Ajouter (un lien) : « Add » pour modifier le modèle d'origine

On peut aussi utiliser Importer : « Import » : pour copier le modèle dans l'espace de travail. La figure suivante montre ces manipulations.

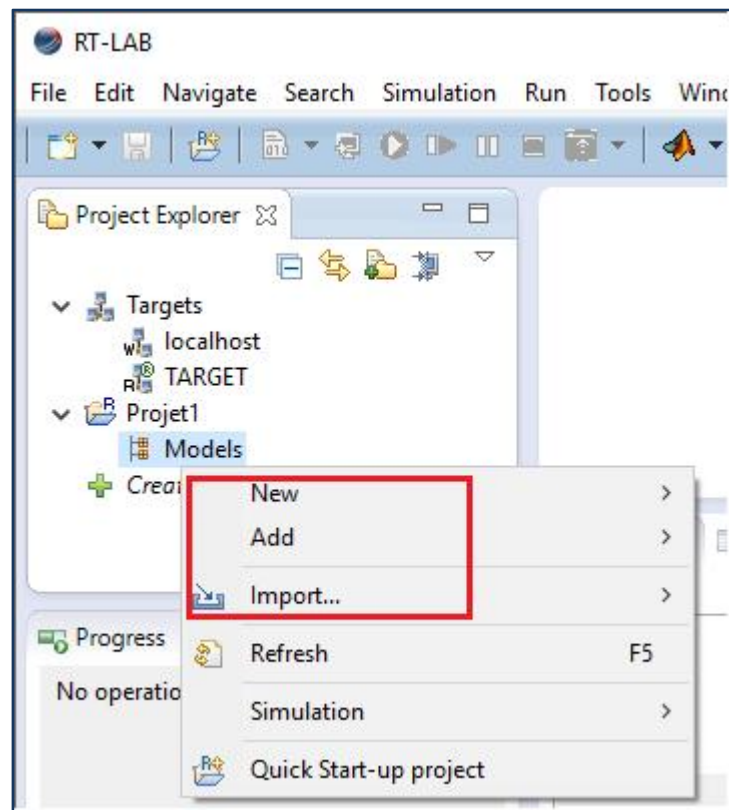


Figure (3.14) : fenêtre pour « Add » ou « Import » des modèles existants

Une fois le projet créé, le modèle additionné ou bien ajouté, on ouvre par défaut le modèle avec la version spécifiée de MATLAB / Simulink. Utilisant : « Edit », le modèle sera éditer et explorer sous une version de Matlab/Simulink choisie. Et le modèle s'ouvre. La figure (3.15) montre plus cette procédure.

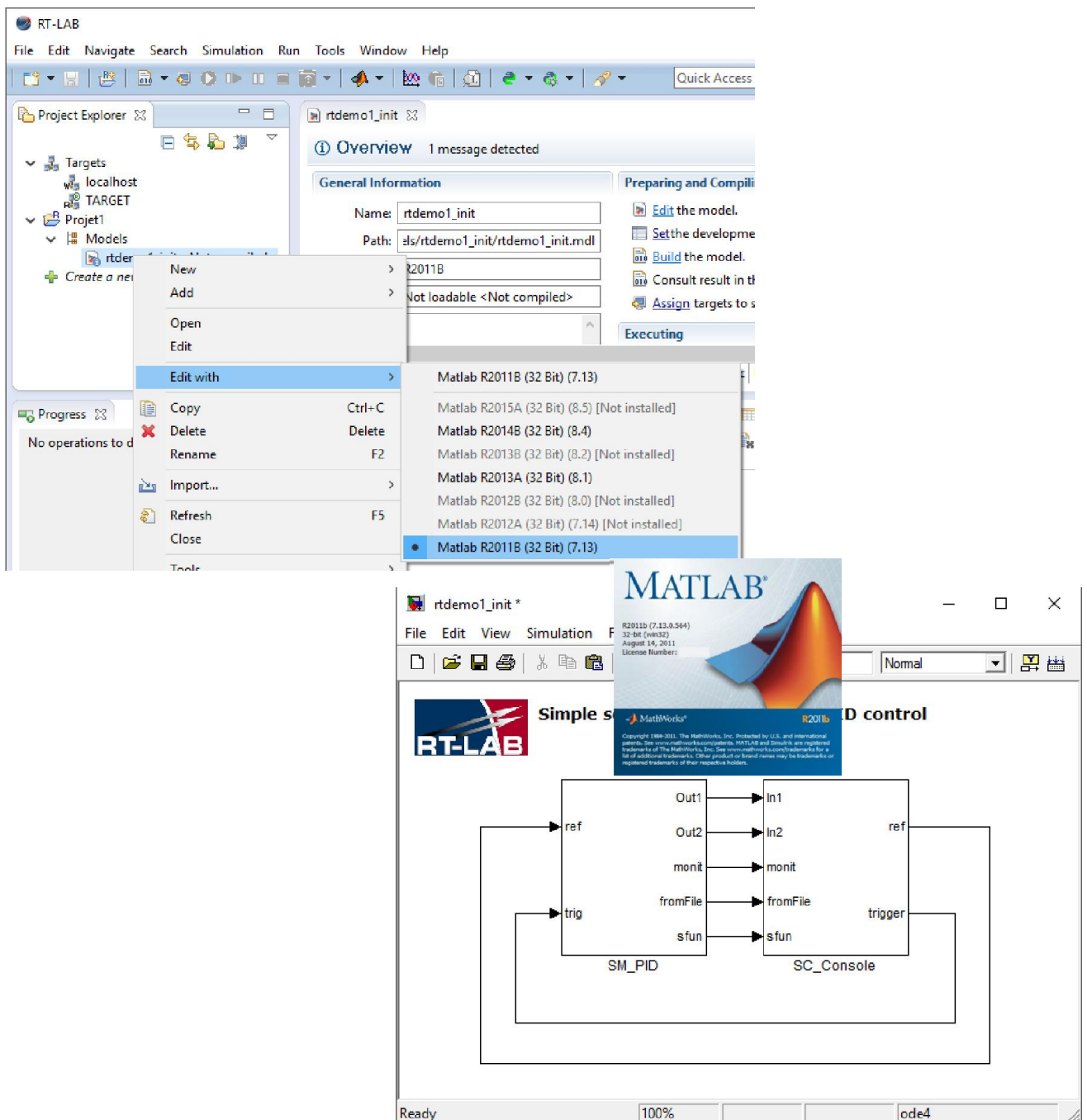


Figure (3.15) : «Edit» et sélection de la version Matlab/Simulink pour ouvrir le modèle

Jusqu'ici le simulateur (Hard) n'est pas opérationnel, on doit établir la liaison entre le PC et le simulateur par le menu ajout de cible « Add a target » soit :

- Manuellement, en utilisant l'adresse IP
 - Utilisation des cibles de découverte (recherche automatique des cibles accessibles).
- « Discover target »

Sachant qu'on a déjà configuré les données réseau des adresses IP, car Le PC « hôte » et la cible doivent être connectés via Ethernet avec une adresse IP compatible.

L'adresse de la cible « target » : « 192.168.10.101 » et celle du PC « 192.168.10.*** » ; valable pour l'OP5410. La figure suivante illustre cette étape.

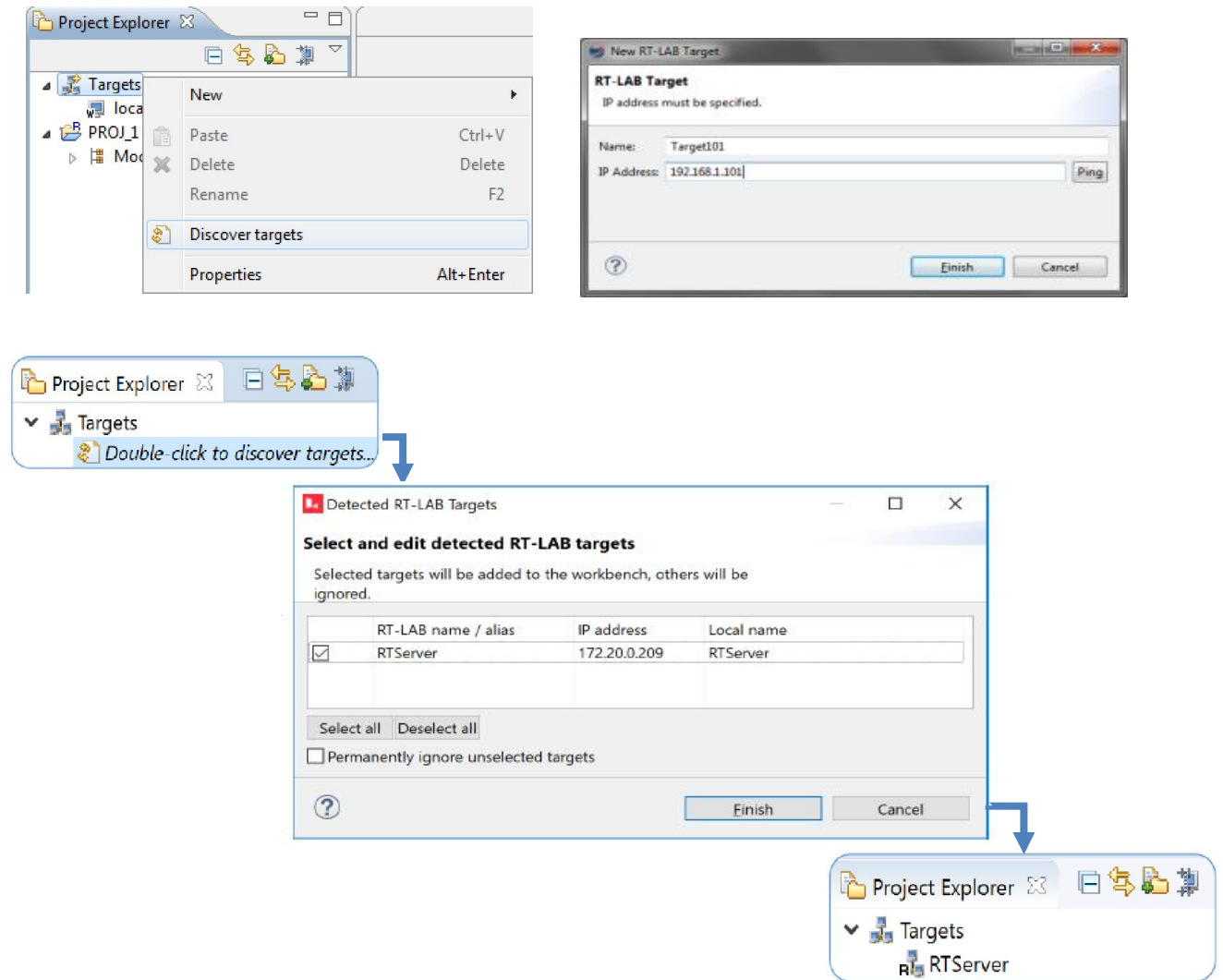


Figure (3. 16) : fenêtres d'activation de la cible « target » PC-Simulateur

3.4 Construction du modèle et son exécution en temps réel

3.4.1 Construction du modèle sous RT-LAB : « Build »

Par sélection du menu « Build », on :

- Sélectionne la plateforme de développement : Versions de logiciels et Plateformes
- Sépare le Modèle : Un fichier .mdl par sous-système de niveau supérieur (SM, SC et SS)

- Génère du code C
- Transférer le code C généré
- Construire le code C généré : Le compilateur cible construit et lie les fichiers à générer un exécutable en temps réel.
- Transférer le modèle construit : Les exécutables sont retransférés sur l'ordinateur.

La figure (3.17) nous montre la phase de construction du modèle avec RT-LAB

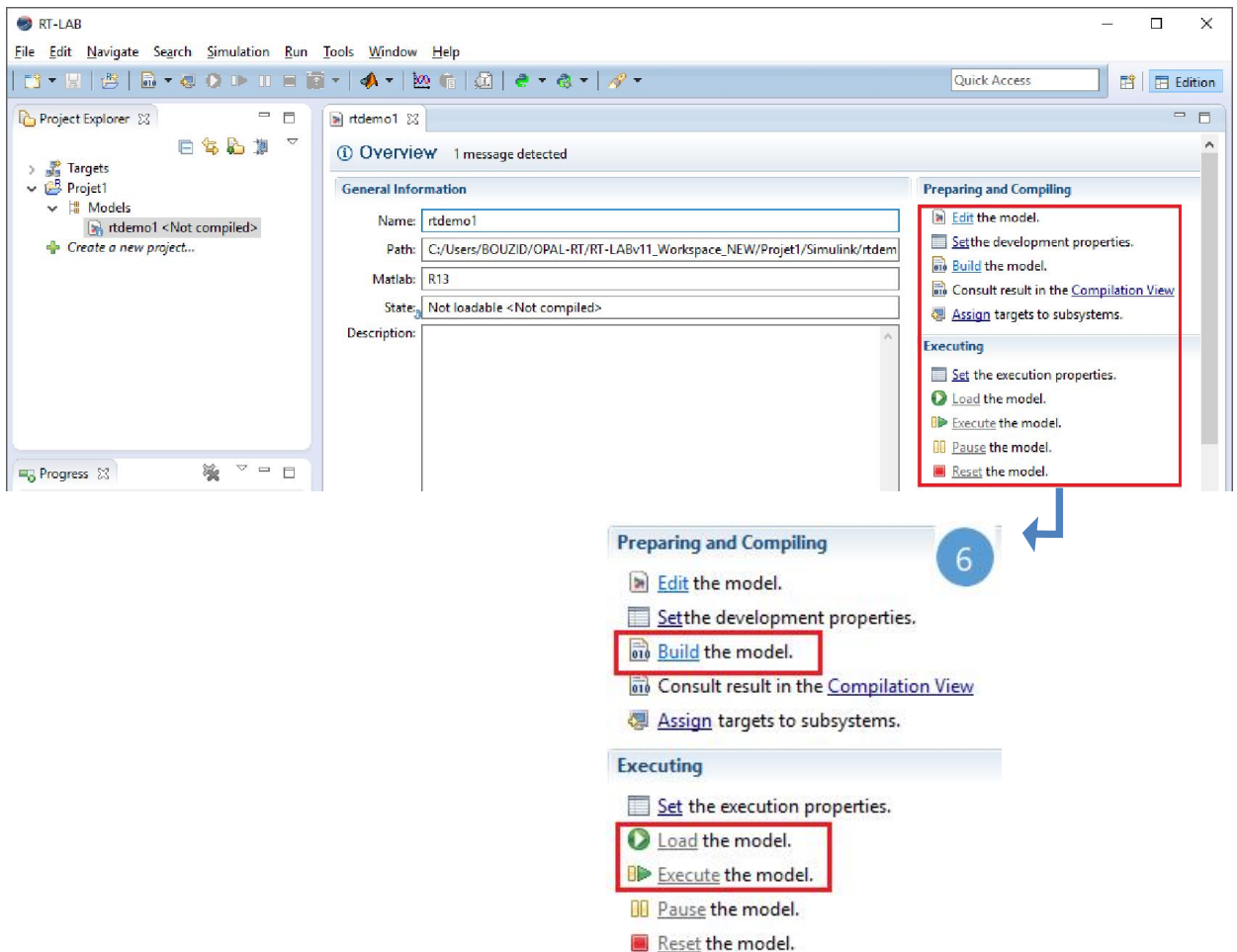


Figure (3.17) : Construction de modèle sous RT-LAB « Build»

3.4.2 Chargé et exécuter le modèle sur la cible

À cette phase on doit sélectionner le mode de simulation soit :

- Simulation
- Logiciel synchronisé (software synchronized)
- Matériel synchronisé (hardware synchronized)

Ensuite on lance le menu « Load » pour Charger le modèle, une fenêtre console apparaît.

Puis on lance « Execute ou Run » pour exécuter le modèle en temps réel sur le simulateur.

Pour finir la simulation en temps réel et afin d'enregistrer des résultats ou Réinitialiser le modèle ou libérer le simulateur, on doit activer « Reset ».

Les figures (3.18) et (3.19) illustrent l'exécution et la réinitialisation du modèle.

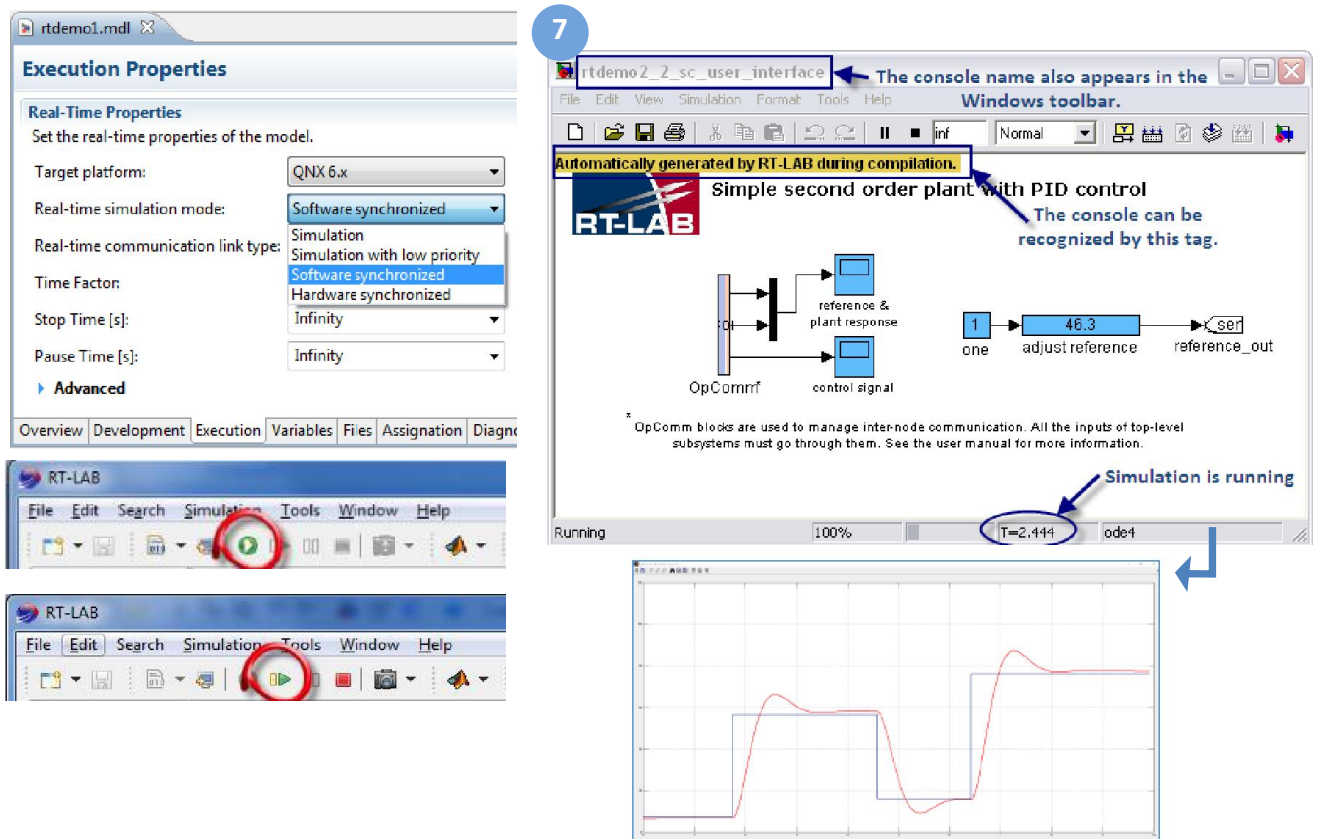


Figure (3.18) : Exécution du modèle et console générée par RT-LAB sous Matlab/Simulink.

La réinitialisation « Reset » arrête la simulation, elle ne redémarre pas le simulateur, après cette étape le modèle est en état « idle »

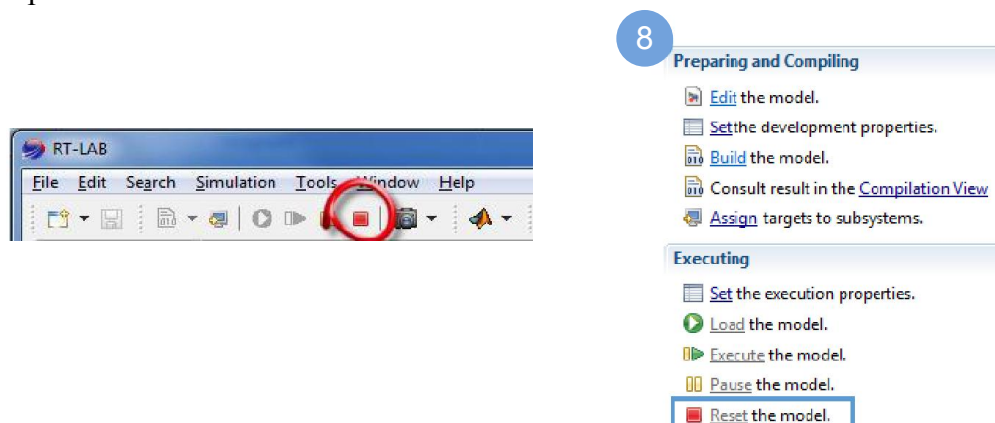


Figure (3.19) : La console Simulink des résultats enregistrés générée après réinitialisation

3.5 Enregistrement de données

RT-LAB permet l'acquisition directe au fichier au lieu de parcourir le réseau vers la console. Cependant le mécanisme sous-jacent est le même. Il y a deux façons de le faire :

- Rediriger les groupes d'acquisition de console (1-25) vers un fichier :
un fichier (myfile_* .mat) sera créé sur chaque nœud de calcul. Ce fichier enregistrera tous les signaux envoyés au groupe d'acquisition sélectionné à partir du sous-système attribué à ce nœud.
- Définissez des groupes d'acquisition dédiés (26-30) qui ne s'écrivent que dans un fichier « write to file » :

Le bloc OpWriteFile de la bibliothèque de blocs RT-LAB de Simulink permet d'enregistrer des signaux dans un fichier. Il contient les mêmes paramètres d'acquisition discutés précédemment. On peut enregistrer plusieurs signaux dans un fichier .mat en utilisant un bloc «Mux» pour créer un vecteur.

3.5.1 Enregistrement de données

On peut avoir jusqu'à cinq blocs « OpWriteFile » dans un modèle. Chaque bloc a un unique nom de fichier entièrement défini par l'utilisateur

Puisque ce bloc permet l'enregistrement de signaux de modèle dans un fichier *.mat. L'utilisateur doit spécifier :

- Nom du fichier *.mat
- Taille maximale du fichier *.mat
- Nom de la variable MATLAB
- Taille de la mémoire tampon d'enregistrement

Le bloc « OpWriteFile » doit être placé dans le sous-système de calcul SM ou SS. Les fichiers générés sont créés dans le simulateur et téléchargés automatiquement sur le PC « host PC ».

La structure de données est exactement la même : La première rangée pour le temps, autres lignes pour les signaux, même ordre d'entrée des signaux dans le bloc.

Ce bloc est également capable d'enregistrer des données hors ligne en cochant l'option « Ecrire dans Mode Simulink : Write to Simulink mode ».

Le fichier « *.mat » est réécrit à chaque fois la simulation est réexécutée.

Les principes d'insertion, de paramétrage et d'enregistrement sont représentés par les figures (3.20) et (3.21).

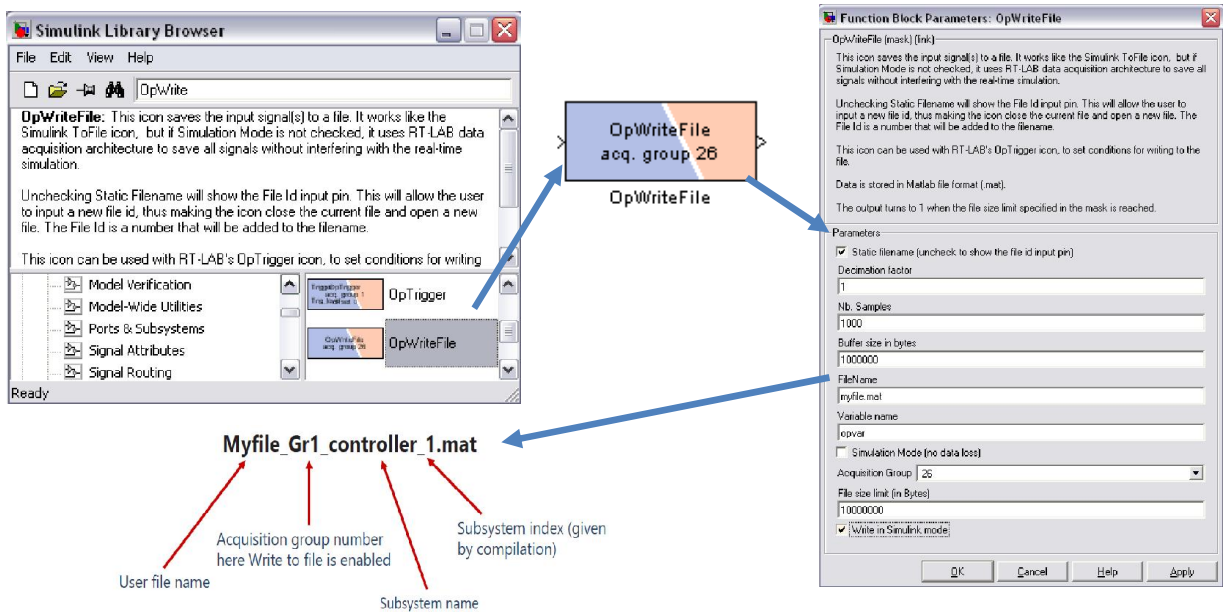


Figure (3.20) : Insertion et paramétrage du bloc « OpWriteFile »

Ce bloc offre la possibilité aussi de :

- enregistrer n'importe quel signal dans le modèle.
- Peut être exécuté même si `sc_console` est fermée.
- Ne bloque pas l'acquisition des paquets TCP / IP
- Besoin de compilation
- Limité à cinq groupes d'acquisition (pas de limitation du fichier .mat cependant)

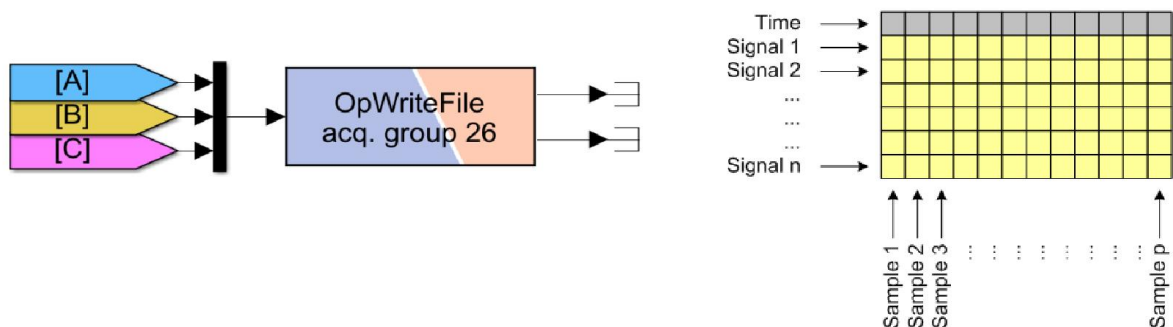


Figure (3.21) : Principe d'enregistrement de données avec « OpWriteFile »

3.5.2 Enregistrement et affichage avec « ScopeView »

ScopeView est une application installée avec RT-LAB, utilisée pour la visualisation des signaux, l'analyse des données et la surveillance. En mode temps réel et hors ligne,

ScopeView affiche, analyse et compare les formes d'onde acquises à partir de divers environnements, systèmes d'acquisition de données et équipements de terrain (physiques).

Scopeview est un outil idéal pour les utilisateurs devant effectuer des calculs mathématiques et électriques, effectuer des analyses de post-traitement et produire des rapports. ScopeView permet à l'utilisateur de se concentrer uniquement sur les résultats, pas sur l'écriture de scripts complexes pour afficher les données.

De plus, ScopeView est disponible avec toutes les packages HYPERSIM , eMEGASIM , eFPGASIM et ePHASORSIM .

3.5.2.1 Caractéristiques principales

Parmi les principales caractéristiques de ce logiciel:

- Visualisez et traitez simultanément les données de nombreuses sources (multi-sources et multi-domaines);
- Exécuter une variété de fonctions mathématiques sur les signaux (par exemple, densité spectrale, réponse en fréquence, cohérence, etc.);
- Offrir une interprétation pour les formules mathématiques qui gèrent les signaux et les scalaires;
- Exécuter de nombreux types de traitements graphiques sur les signaux (ex: zoom, superposition, versus, curseur de suivi, déplacement interactif de graphiques, etc.);
- Enregistrer les informations de traitement et d'édition du signal dans un fichier d'acquisition de modèle pour une utilisation ultérieure;
- Permet d'exporter les signaux acquis dans différents formats (par exemple MATLAB, ASCII, PDF, POST-SCRIPT)

3.6 Conclusion

Avant de lancer la simulation en temps réel avec RT-LAB , et avoir une acquisition de données et des résultats on doit préparer le modèle à simuler selon des critères propres et spécifiques à (RT-LAB).

Une fois RT-LAB installé, une boîte d'outils s'intègre à la bibliothèque Simulink (RT- LAB, RT-EVENTS, RT-LAB dynamo, RT-LAB I/O, RT-LAB Drive, Artemis, eFPGAsim , eHS, MMC etc..). Ce qui permet l'accès au modèle en matlab/Simulink via l'interface RT- LAB.

Le groupement de tous les blocs du modèle en sous systèmes ; un sous système de calcul et un sous système de l'interface console (SM et SC) et l'ajout des blocs de communication sont nécessaires pour établir la liaison TCP/IP entre le PC (Host) et la cible (Simulateur).

Ensuite on doit respecter les étapes pour éditer, compiler et exécuter le modèle.

Avec des préréglages selon le modèle à simuler, RT-LAB offre la possibilité de contrôler, surveiller et de visualiser les variables et les paramètres pendant la simulation (online). On peut aussi enregistrer et comparer des résultats numériques ou graphiques à l'aide des outils (blocs) disponibles en boîte à outils Simulink (RT-LAB tools) ; tel que OpWritefile, OpMonitor et ScopeView.

CHAPITRE 4

*APPLICATIONS SUR UN SYSTEME
D'ENTRAINEMENT ELECTRIQUE D'UNE MSAP*

4.1 Introduction

Après avoir traité les principes de base de la simulation en temps réel avec RT-LAB et les différentes fonctionnalités, configurations et paramétrages nécessaires, on aborde dans cette partie des simulations réelles réalisées avec OP4510 et Matlab R2017a sur un modèle d'entraînement d'une machine synchrone à aimants permanents alimenté par convertisseur électronique employé pour la traction d'un train électrique (High-power PMSM train drive with 3-level NPC inverter and 12-pulse thyristor rectifier).[04]

Ce modèle met en œuvre un entraînement de train « PMSM » haute puissance avec un onduleur NPC à 3 niveaux et un redresseur à thyristor à 12 impulsions.

Le redresseur est modélisé avec SSN et n'a pas de retard de modélisation entre le transformateur et les redresseurs.

Le moteur « PMSM » est évalué à 1 MW et est piloté par un onduleur modèle NPC-TSB à 3 niveaux avec une capacité à haute impédance. Il s'agit d'un modèle ADDRIVE-01 mis à jour utilisant le SSN et le TSB à haute impédance.

Une description du circuit est donnée en premier, décrivant les principaux composants du circuit, à savoir, la topologie du pont convertisseur, les niveaux de tension et les formes d'onde, le générateur PWM, le processus de filtrage, la charge, les mesures et enfin le comportement du moteur à savoir la vitesse de rotation et le couple mécanique.

4.2 Description générale du système (circuit) [04], [05], [06]

Dans le schéma de la figure (4.1), il excite deux parties (partie de puissance CPU1 ($T_s : 50 \mu s$) et partie de commande CPU2 ($T_s : 25 \mu s$)), pour définir chaque partie il faut savoir la constitution de la partie :

CPU1 : circuit de puissance

- Générateur d'alimentation 3Ø (20 kV)
- Transformateur 3Ø avec couplage Y-YΔ (1,5 MVA, 50Hz), (20Kv-1kV)
- Convertisseur (AC/DC) (redresseur) avec 12 thyristors avec deux bras
- Batterie de condensateur (4mH)

Donc l'utilité de CPU1 c'est l'alimentation de CPU2 avec la charge limite des batteries de 12 μF .

La présence de L (4mH) c'est une bobine de lissage qui permette d'éliminer les harmonique ou bien filtrer le courant de charge.

CPU2 : circuit de commande

- Convertisseur (DC/AC) à 3 niveaux avec deux bars
- Moteur synchrone à aimants permanents (PMSM - 1MW)

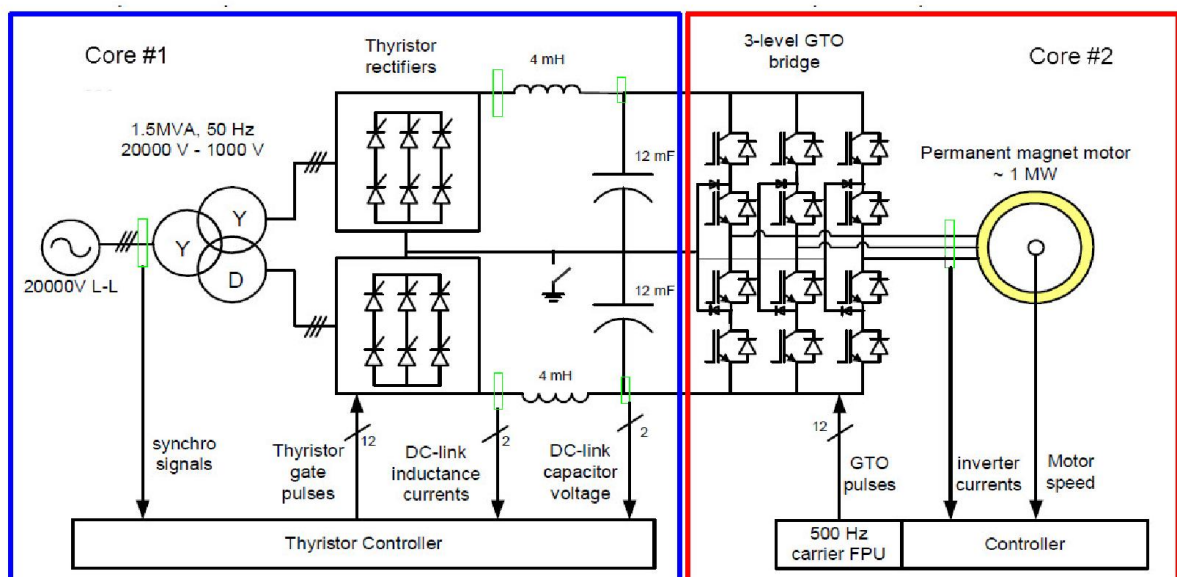


Figure (4. 1) : Circuit des différents éléments du modèle étudié [04], [05], [06]

Le convertisseur « convertir DC/AC » est contrôlé par une loi de commande pour alimenter l'induit de moteur. Afin de surveiller et contrôler le couple et la vitesse en temps réel

L'entraînement de traction du train décrit à la figure 4.1 est composé d'un redresseur à thyristor à 12 impulsions connecté au réseau et connecté à un onduleur GTO/IGBT à 3 niveaux alimentant un moteur à aimants permanents de 1 MW.

L'onduleur GTO à 3 niveaux est modélisé avec un pont horodaté (TSB) et ARTEMIS a été utilisé pour réaliser une simulation dure en temps réel du redresseur côté AC.

De plus, un modèle de transformateur spécial de la bibliothèque de modèles en temps réel d'Opal-RT est utilisé pour découpler artificiellement les deux enroulements secondaires du transformateur afin que le précalcul complet des deux ponts de thyristor à 6 impulsions puisse être effectué par ARTEMIS. Sans cela, l'algorithme doit précalculer $4096 (2^{12})$ équations système différentes pour le côté AC/DC uniquement [2], [5].

Le principe utilisé pour introduire le découplage au niveau des enroulements secondaires est de modéliser l'inductance de fuite secondaire par une ligne de transmission courte (parfois appelée « stubline ») ayant la même inductance de ligne. L'approximation a alors pour effet d'introduire des capacités de ligne qui ne sont pas réellement présentes dans le circuit. Tant que le temps d'échantillonnage est petit, les capacités parasites sont également petites et l'erreur est minime.

Avec un circuit intermédiaire connecté à un onduleur GTO (GateTurn-Off thyristor) à 3 niveaux alimentant un moteur à aimants permanents 1 MW, pour réaliser une simulation en temps réel dur du redresseur côté AC, l'onduleur a été modélisé avec les ponts horodatés (STB) de ARTEMiS.

La compensation de l'onduleur est effectuée dans le solveur SSN, sans aucun retard, et la machine « PMSM » est modélisée en utilisant la méthodologie SSN. ca ne nécessite notamment pas que les amortisseurs soient précis et stables en modes haute impédance et redressement. Il s'agit d'une variante qui utilisait le TSB (modèle à fonction de commutation d'interpolation) pour l'onduleur, avec le « PWM » piloté jusqu'à 9 kHz avec des effets de temps mort.

L'équipement et la configuration pour la simulation en temps réel sont décrits ci-après, montrant les plates-formes matérielle et logicielle utilisées, suivi d'une description détaillée des différentes étapes.

4.3 Entraînement de la machine synchrone à aimants permanents (PMSM drive) [07], [08]

Les principaux marchés des PMSM sont l'industrie automobile et des transports. Ce type de machine est généralement choisi en raison de son excellente densité de puissance (puissance produite par rapport à la taille ou au poids) ou de sa capacité à atteindre une vitesse plus élevée que les autres types de moteurs ainsi que des couples importants.

Cependant, contrôler un moteur « PMSM » est généralement plus difficile que d'autres types de machines. Puisqu'il s'agit d'une machine synchrone, le contrôleur doit être conscient de la position du rotor à tout moment afin de contrôler correctement le couple. De plus, il existe un risque de décantation de l'aimant si la commande n'est pas stable, ce qui entraînera une modification des propriétés de la machine.

L'exemple qu'on a choisi décrit la mise en œuvre sur le simulateur temps réel RT-LAB d'un variateur à moteur synchrone à aimant permanent (PMSM) avec un pas de temps de simulation de 50 μ s. Le variateur utilisé est constitué d'un onduleur de tension à GTO; le circuit intermédiaire est constitué d'un condensateur chargé par deux ponts redresseurs, chaque pont à 6 diodes connecté au réseau par l'intermédiaire d'un transformateur à deux enroulements secondaires,

On représente dans cette partie les techniques de simulation du modèle d'entraînement de la machine synchrone à aimants permanents en temps réel sur le simulateur OP4510 et l'emploi des packages machines électriques et convertisseurs statiques développés par OPAL-RT. Ils sont spécifiquement conçus pour étudier le fonctionnement, réaliser des essais et la mise en service des entraînements des machines électriques à savoir l'application dans la traction électrique.

4.3.1 Intérêt de La machine synchrone à aimants permanents « PMSM »

Le terme de la machine synchrone regroupe toutes les machines dont la vitesse de rotation de l'arbre de sortie est égale à la vitesse de rotation du champ tournant. Pour obtenir un tel fonctionnement, le champ magnétique rotorique est généré soit par des aimants, soit par un circuit d'excitation.

Le moteur synchrone est constitué de deux parties, partie fixe ou stator soutenant des enroulements constituant l'induit, et une partie mobile ou rotor constituant l'inducteur.

Le rotor ou roue polaire, est réalisé par un circuit magnétique comportant des aimants permanents.

La position du champ rotorique est alors fixe par rapport au rotor, ce qui impose en fonctionnement normal une vitesse de rotation identique entre le rotor et le champ tournant statorique.

Le fonctionnement de la machine synchrone à aimant permanent est beaucoup plus simple que celui du moteur à courant continu : le stator muni d'un enroulement polyphasé, triphasé dans la plus part des cas, est alimenté par un système de tensions et courants créant dans l'entrefer un champ d'induction tournante. *Bri*, le champ d'induction *Bri* a tendance à attirer le rotor, lequel est muni d'aimants permanents produisant le champ d'induction. De ce fait, les champs d'induction créés par le stator et le rotor ont tendance à s'aligner, raison pour laquelle un couple d'origine électromagnétique prend naissance *Bri*. Comparativement à la machine à courant continu, on voit que c'est le rotor qui joue le rôle d'inducteur (excitation), l'induit étant alors au stator. Le rotor se met à tourner à une vitesse angulaire ω correspondant à la pulsation ω de l'alimentation triphasée, ce que justifie la désignation du moteur synchrone.

4.3.1.1 Modèles du moteur (PMSM) disponibles

L'étude de tout système physique nécessite une modélisation Celle-ci nous permet de simuler le comportement de ce système face à différentes sollicitations et d'appréhender ainsi le mécanisme régissant son fonctionnement. Les lois modernes de commande de plus en plus performantes permettent un meilleur contrôle des régimes transitoires tout en assurant, dans une grande plage de fonctionnement, un asservissement précis de la vitesse, tous ces perfectionnements demandent une bonne connaissance de la machine et de son convertisseur. Dans notre cas, nous en déduirons des lois de conception adaptées aux machines synchrones à aimants permanents.

Selon le solveur (RT-LAB / Solveur PMSM) trois types de modèles de moteur sont disponibles pour toutes les configurations FPGA-PMSM: modèle DQ standard sans saturation, modèle Variable-DQ (VDQ) et modèle d'harmonique spatiale (SH). Les caractéristiques de mise en équation du moteur dans le repère de D-Q standard seront prises dans cet exemple.

4.3.2 Description du bloc convertisseur à trois niveaux (3level TSB-RTE Drive) [02], [09]

Le bloc Pont convertisseur horodaté à 3 niveaux (3L-TSB) RTE-Drive implémente un pont à 3 niveaux (3 commutateurs) dont les portes sont contrôlées par des signaux booléens RTE. Le nombre de bras en parallèle peut être choisi de 1 à 3. Le pont modélise des dispositifs IGBT/GTO/MOSFET contrôlés par un interrupteur idéal avec leurs diodes

anti-parallèle (RD). La figure suivante présente le circuit électrique équivalent du bloc RTE-Drive TSB 3-Niveaux à 3 bras.

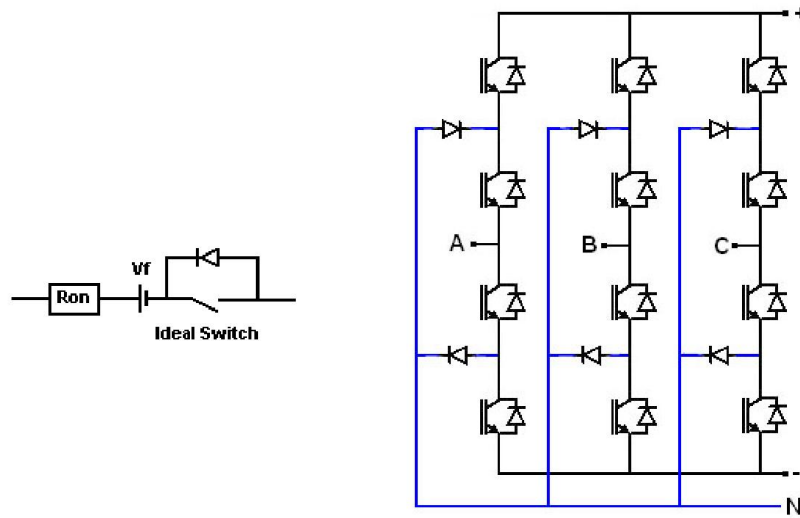


Figure (4. 2) : Circuit électrique équivalent du convertisseur TSB 3-Niveaux – RTE-DRIVE

Le modèle de commutateur est simulé comme une résistance R_{on} et une source de tension continue V_f , connectées en série avec un commutateur. L'interrupteur est commandé par le signal de gâchette (G).

Lorsque l'interrupteur supérieur (ou la diode anti-parallèle supérieure) conduit, la sortie V_{abc} est égale à l'entrée (V^+) « moins les chutes de tension internes ». Lorsque l'interrupteur ou la diode inférieure du bras conduit, la sortie V_{abc} est égale à (V^-) « plus les chutes de tension internes ». Cette charge, qui a une causalité tension-entrée courant-sortie (comme une inductance), produit un courant qui est fourni à l'entrée I_{abc} .

La sortie (I^+) est calculée à partir de tous les temps de conduction des interrupteurs ou diodes supérieurs. La sortie (I^-) est calculée à partir de tous les temps de conduction inférieurs des commutateurs ou des diodes.

Le bloc n'est pas précis lors de la simulation de l'effet de la coupure complète de toutes les impulsions de porte *IGBT*, comme lors d'un arrêt d'urgence par exemple. Cela a pour effet de laisser la diode anti-parallèle s'allumer/éteindre naturellement, un cas pour lequel le pont horodaté n'est pas conçu.

Depuis la version 3.3.x de RT-EVENTS, des optimisations dans les ponts d'horodatage ont été effectuées pour réduire le délai d'entrée-sortie dû à l'interfaçage Simulink-SPS. Cette optimisation concerne le *TSB* lorsque le paramètre Output Data Type est SimPowerSystems.

Étant donné que le pont d'horodatage a une fonction de commutation pour laquelle les sorties dépendent directement des entrées, les boucles algébriques doivent être rompues lors de la connexion en mode SimPowerSystems, ce qui est obtenu en forçant des retards temporels uniques à l'entrée(s) ou les sorties du modèle.

L'optimisation implique le déplacement des retards de simulation de la ou des tension(s) alternative(s) et courant(s) continu(s) de sortie vers la ou les tension(s) continue(s) et courant alternatif d'entrée. Cette modification permet de mieux simuler une forte majorité d'applications où l'on retrouve une tension constante sur le(s) bus continu (ou la constante de temps de la tension continue est faible par rapport au pas de temps), du fait d'une source de tension fixe ou actions de contrôle.

La figure (4.3) montre le résultat de l'analyse du cycle d'utilisation pour la technique d'horodatage (*TSB*) par rapport à la technique de blocs SimPowerSystem. Si les deux techniques ont la même réponse au pas de temps de simulation de 0,5 μ s, elles diffèrent beaucoup à 50 μ s : le SimPowerSystem présente une forte non-linéarité (courbe rouge) alors que le modèle de pont horodaté est toujours linéaire (courbe bleue). Le pont horodaté est aussi précis que les simulations de référence faites à 0,5 μ s.

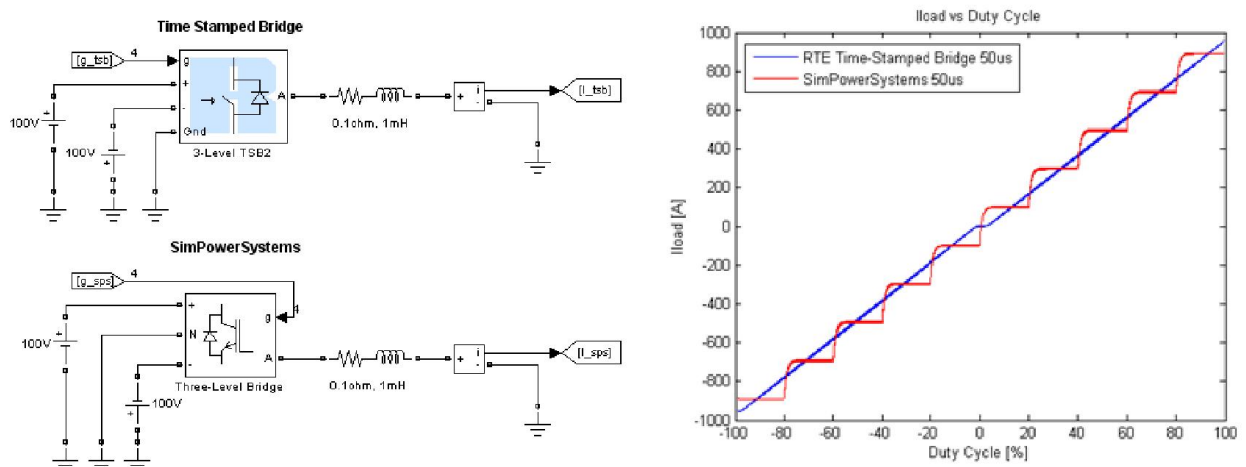


Figure (4. 3) : Analyse du cycle d'utilisation pour la technique d'horodatage (TSB) RTE-DRIVE

4.4 Simulation du modèle « TRAIN TRACTION DRIVE » sur L'OP4510 [04], [05]

Cette démonstration met en œuvre un moteur synchrone à aimants permanents alimenté par deux redresseurs à thyristors triphasés. De l'autre côté, le moteur à aimants permanents est entraîné par un onduleur à *GTO* à commande *PWM* (**3phase 3level GTO/IGBT inverter drive**). Le bus continu (**DC-link**) est alimenté par une source triphasée via les deux

redresseurs à thyristors (**12-Pulse Thyristor Rectifier AC-Side**). Le temps mort de l'onduleur est pris en compte dans le modèle de simulation.

Divers phénomènes, tels que les caractéristiques des convertisseurs AC-DC et DC-AC; entraînement du moteur (PMSM) à faible vitesse et fort couple; effets des contrôles d'allumage du convertisseur et du temps mort (Dead time), effet des harmoniques de tension; peuvent être étudiées et validées avec ce modèle.

À noter que vu la grande puissance ($1 \div 1,5$ Mw) et le couple important de notre application pour la traction d'un train, le moteur sera entraîné à faible vitesse et fort couple (460 Nm en régime normal). afin de vaincre la forte inertie. Ensuite on réalise un test avec l'addition d'un couple de charge supplémentaire de l'ordre de 1000 Nm après que le régime permanent soit établi.

Cette simulation en temps réel a fonctionné sur un « OP4510 - dual-Xeon-based, 2.33 GHz » et RT LAB2019 / Matlab r2017a, plus de détails figurent au tableau (4.1). [05].

	CPU 1: Redresseur (50 μs)	CPU 2: Onduleur (25 μs)
Contenu des composants	• 3 sources de tensions contrôlées • 1 Générateur d'harmonique de tension. • 1 Transformateur Y/Y-Δ (avec stublines). • 1 Contrôleur du redresseur • 2 redresseurs à six thyristors • 2 Circuit intermédiaire DC (de condensateurs) + découplage	• 1 Onduleur GTO à 3 niveaux (TSB) • 1 MSAP (PMSM) • 1 Contrôleur de l'onduleur
Temps de calcul du modèle.	7 μ s (14%)	6 μ s (12%)
Pas de temps minimum	8 μ s	
Facteur d'accélération	5 (50 μ s)	

Tableau (4. 1) : Allocation de processeur et performances en temps réel. [05]

4.4.1 Étapes de déroulement de la simulation en temps réel [02]

Les figures suivantes (4.5), (4.6) montrent la création le lancement et l'exploration du modèle Matlab/simulink à partir de RT-LAB

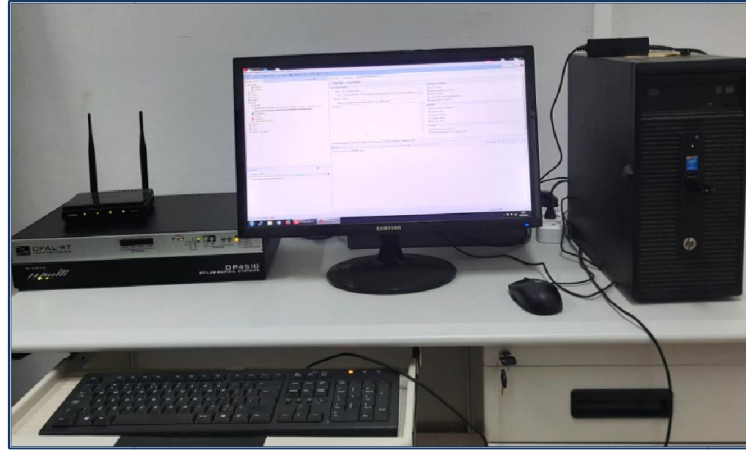


Figure (4. 4) : Simulateur OP4510 relié au PC (du laboratoire LGE)

le modèle RT-LAB correspondant sera importé dans l'espace de travail du projet (dans cet exemple, le modèle est nommé (DRIVE01_TRAINTRACTION).

Après édition du modèle avec MATLAB/simulink, on remarque que ce modèle se compose de trois sous systèmes : un sous système d'affichage « SC_UserInterface » et deux sous systèmes de calcul le premier « SM_PMSM_Drive » le deuxième « SS_AC_Rectifier » ce qui nécessite l'occupation de deux cœurs (CPU) de calcul au niveau du simulateur. Le sous système « SS_AC_Rectifier » ou (slave) a pour tâche le calcul du contrôle de la partie AC/DC alimentant la partie d'entraînement ou (master) « SM_PMSM_Drive ». t le sous système « SM_PMSM_Drive » assure la commande des impulsions de l'onduleur (PWM génération) et le contrôle des courants de la vitesse et du couple du moteur « PMSM »

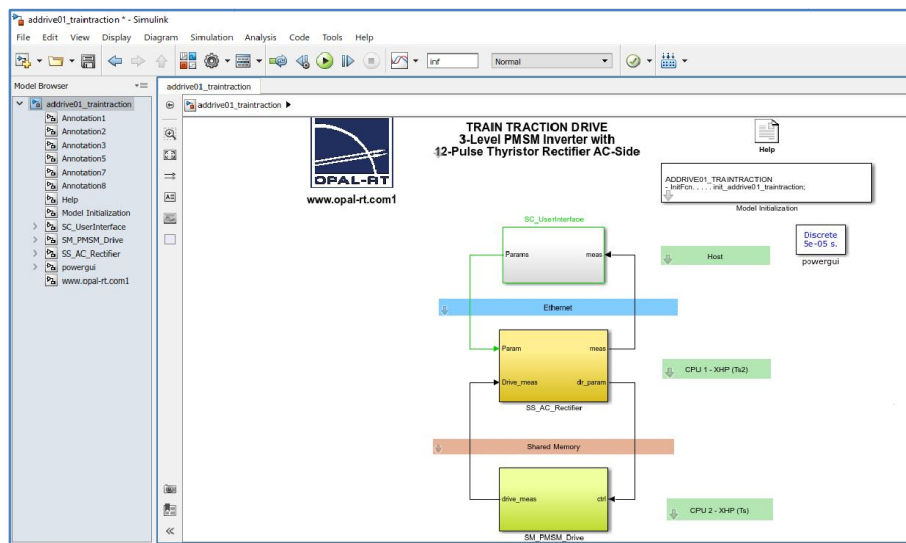


Figure (4.5-a) : L'Édition du modèle étudié avec RT-LAB/Matlab-Simulink

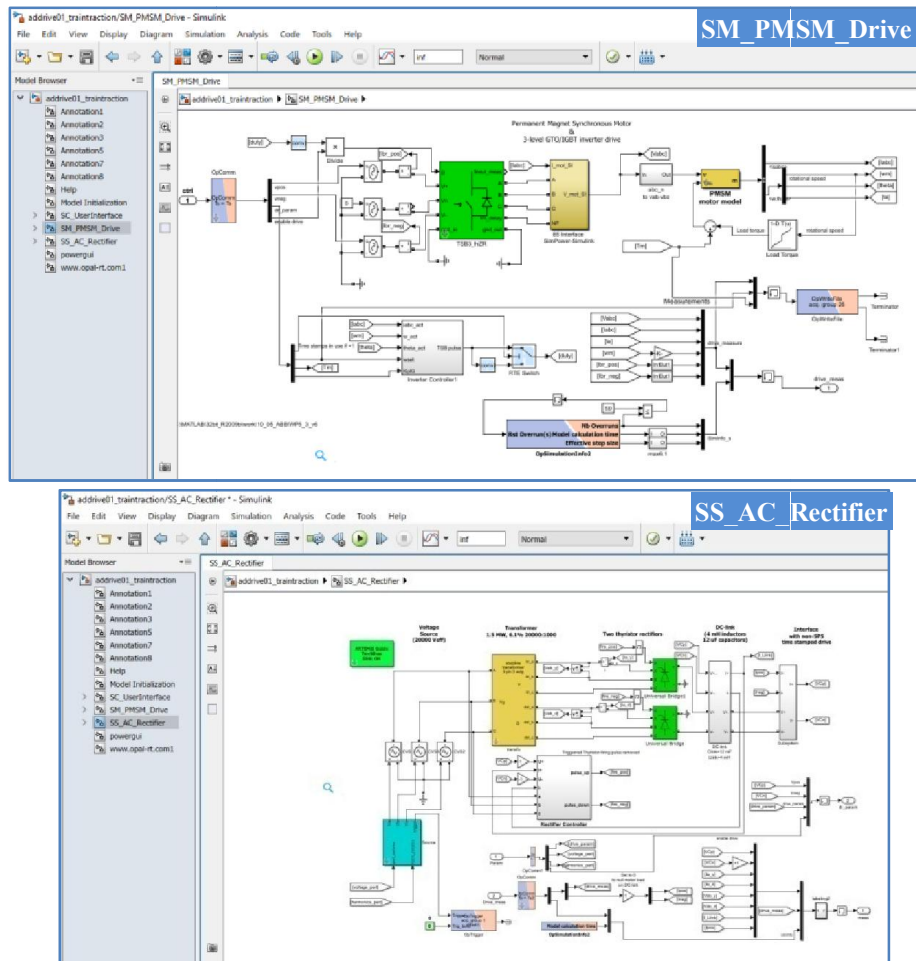


Figure (4.5-b) : Les deux sous systèmes du modèle édité

Le « **build** » lance une construction du modèle et le compile avec des langages et codes spécifiques. La figure (4.6) illustre l'opération.

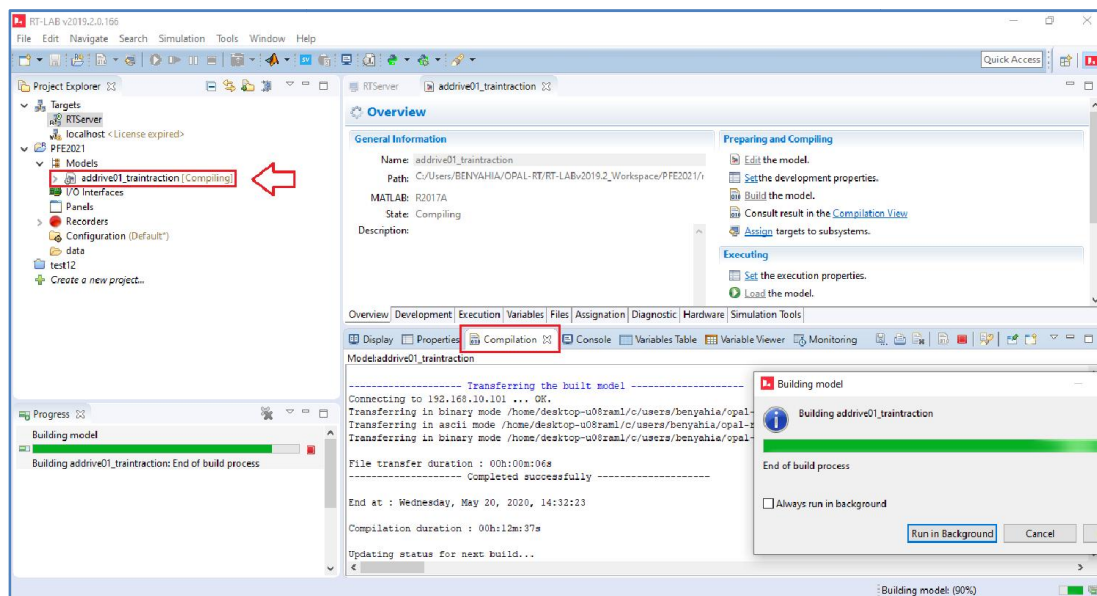


Figure (4.6): La construction et la compilation du modèle

Il est indispensable avant la compilation du modèle « **build** », d'attribuer la cible (target) du simulateur afin d'établir la communication avec le simulateur ; ici « RTServer ». La licence sera vérifiée avec le système.

Avec le menu (Assignation) on vérifie l'attribution des cœurs (deux pour notre cas) pour chaque sous système de calcul, ainsi que le mode XHP (eXtra High Performances) doit être activé dans ce cas afin d'assurer une simulation en temps réel performante.

Dans l'onglet (EXECUTION PROPERTIES), on définit le mode de simulation en temps réel sur Software Synchronized.

Dans la figure (4.7), on remarque bien que le mode XHP est activé, et deux cœurs du CPU de « RTServer » sont activés.

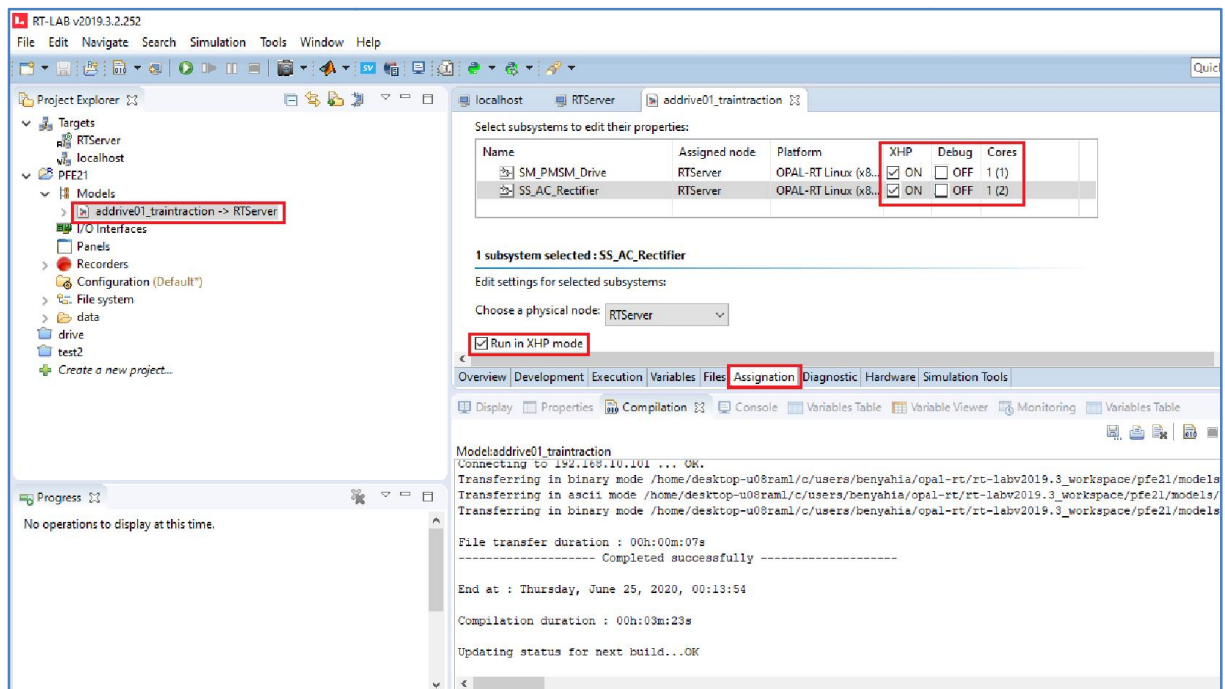


Figure (4.7): L'assignation et l'activation du mode XHP

Après la compilation du modèle avec succès (sans erreurs), maintenant notre le modèle est prêt à être chargé et exécuté sur le simulateur, pour ce faire dans l'onglet de Présentation, on active « **Load** ». à la fin de cette opération une console d'affichage s'ouvrira, et le code en temps réel sera téléchargé au simulateur. La fenêtre de "The Loading model " apparaît brièvement pendant le processus de chargement (Load).

Le chargement du modèle ainsi que la fenêtre de la console d'affichage graphique générée, sont montrés sur les figures (4.8) et (4.9).

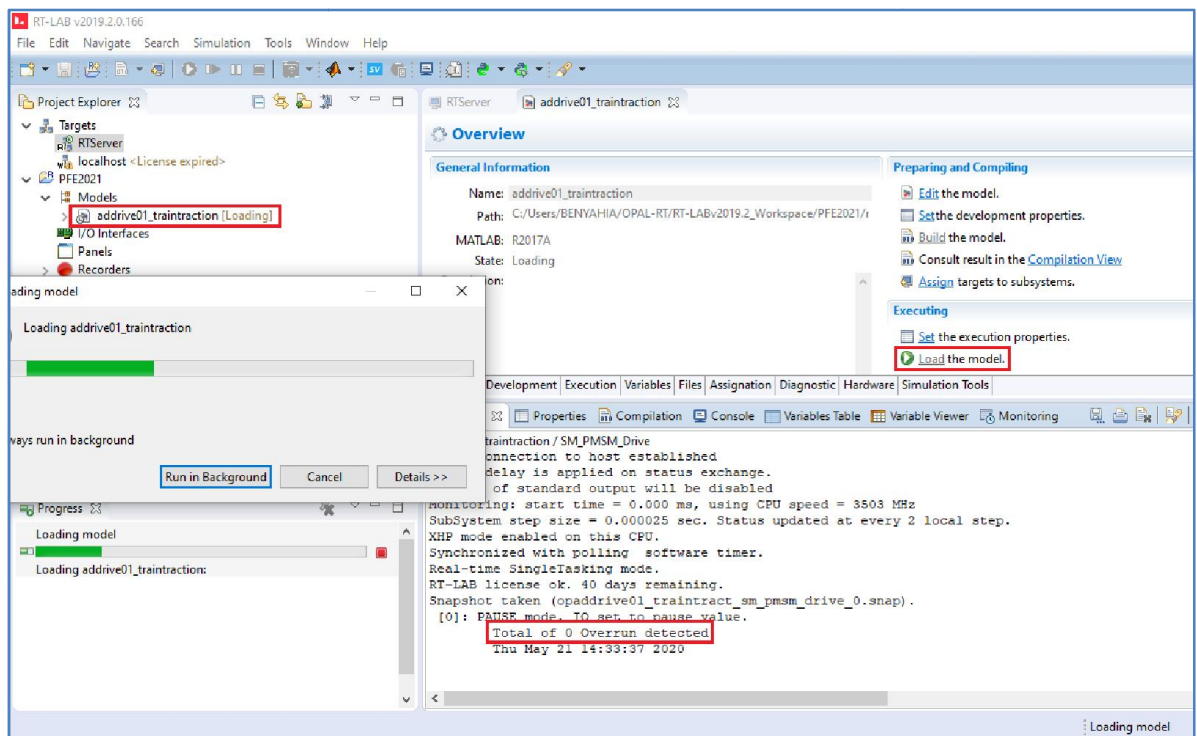


Figure (4.8): Chargement « load » du modèle au simulateur

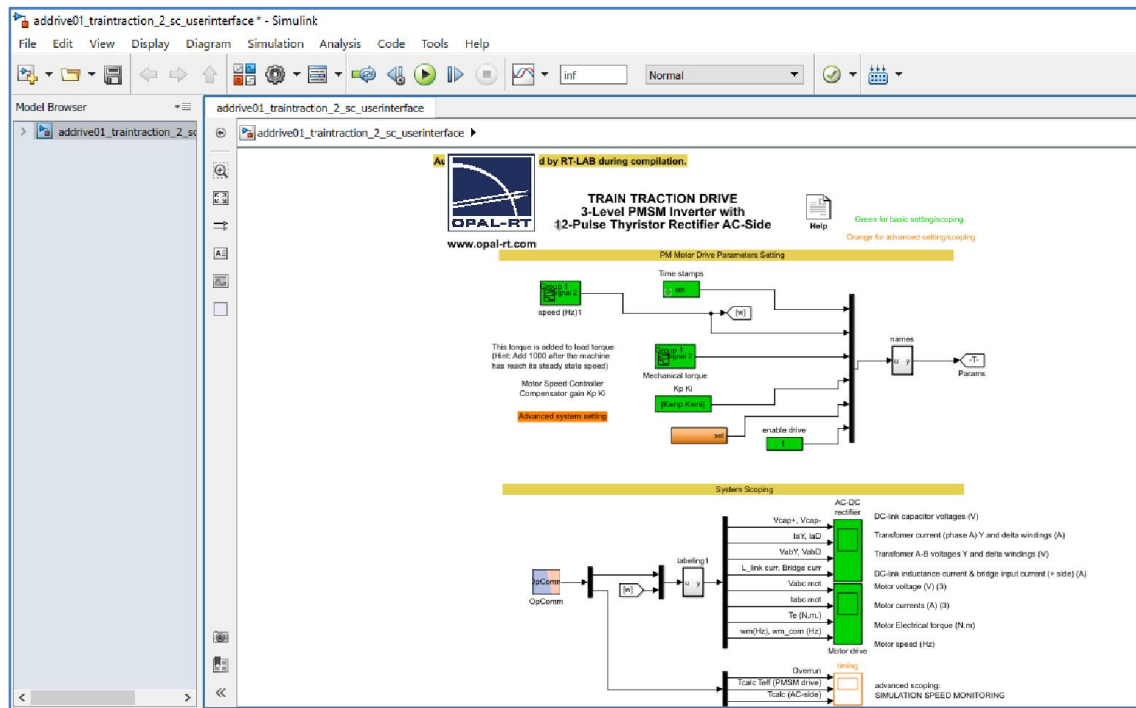


Figure (4.9): La console d'affichage générée après chargement du modèle

L'onglet d'Affichage (Display tab) dans la partie inférieure de la fenêtre affichera la progression et les détails du chargement.

L'onglet **Monitoring** affiche l'état de calcul en temps réel ainsi les interruptions ou dépassements dans la simulation pour les deux sous systèmes de calcul.

On activant « exécute /run » notre modèle s'exécute en temps réel sur le simulateur, depuis la fenêtre développée de la console Simulink, on peut de visualiser surveiller contrôler et afficher des mesure de la simulation en cours d'exécution dans la fenêtre de la console.

La figure suivante montre la fenêtre de l'interface RT-LAB indiquant que le modèle s'exécute sur le simulateur avec affichage du message pas de dépassements (0 overruns). On remarque au bas d'affichage un fichier myfiletrain.mat est créé.

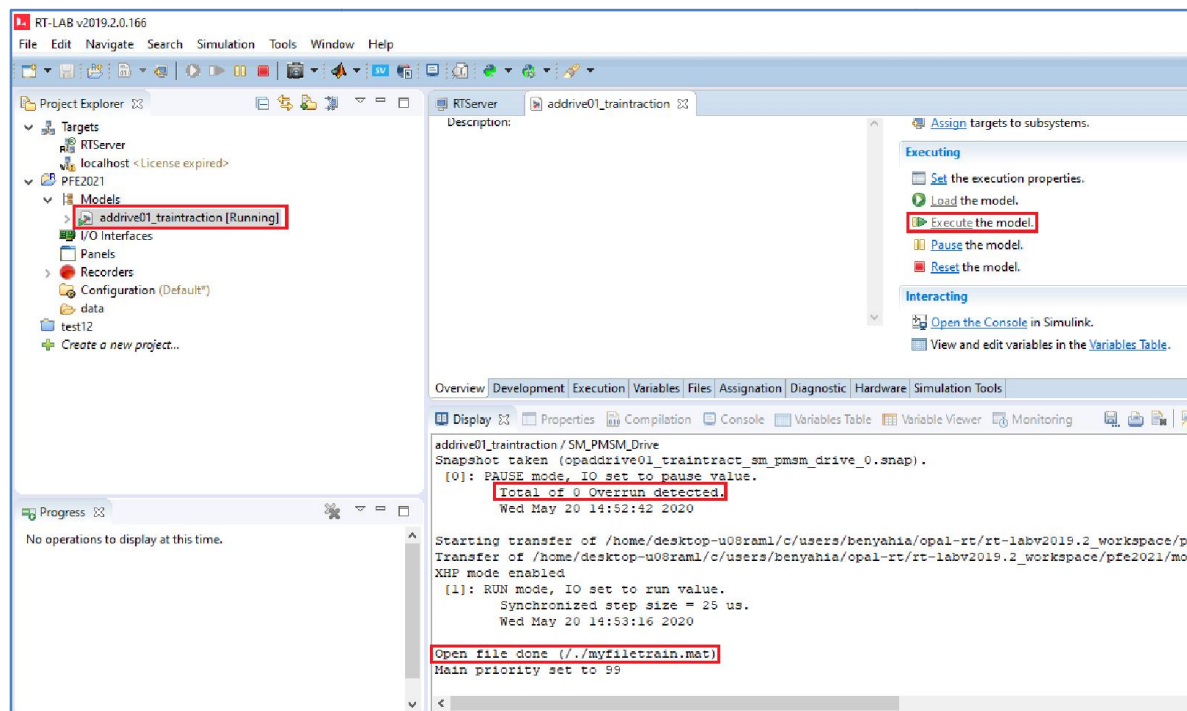


Figure (4.10): phase d'Exécution du modèle sur le simulateur en temps réel

Une action sur « reset » nous permet arrête la simulation en temps réel, réinitialise le model et libère le simulateur. A ce moment on peut afficher et vérifier les enregistrements déjà lancés au cours de la simulation en temps réel.

Sur la figure (4.11), on remarque notre modèle et réinitialiser, et RT-LAB lance la récupération de fichiers tel que le fichier « myfiletrain.mat » dont le quel on a enregistré nos résultats (graphiques) via le bloc « Opwritefile ».

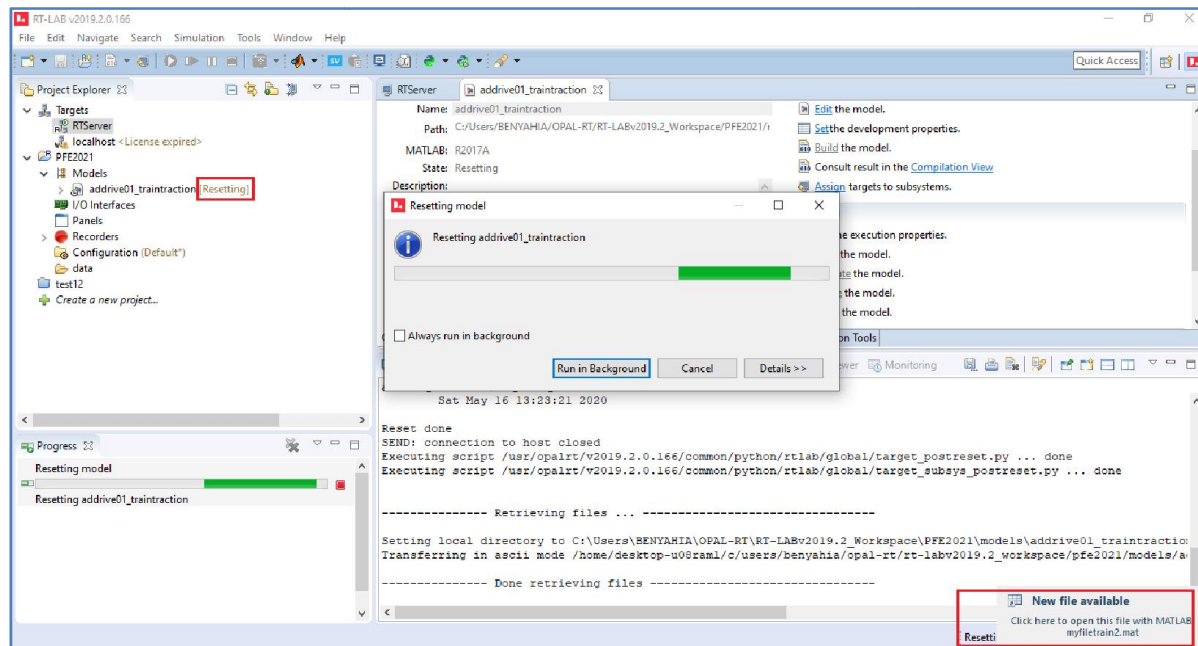


Figure (4.11): réinitialisation de la simulation « reset »

Afin de vérifier le déroulement et les performances de la simulation de notre modèle en temps réel, on a activé l'affichage du « monitoring » durant la phase de l'exécution du modèle avec XHP activé et avec désactivation du mode XHP. Les figures qui suivent (4.12-a), (4.12-b) et (4.13) montrent la différence.

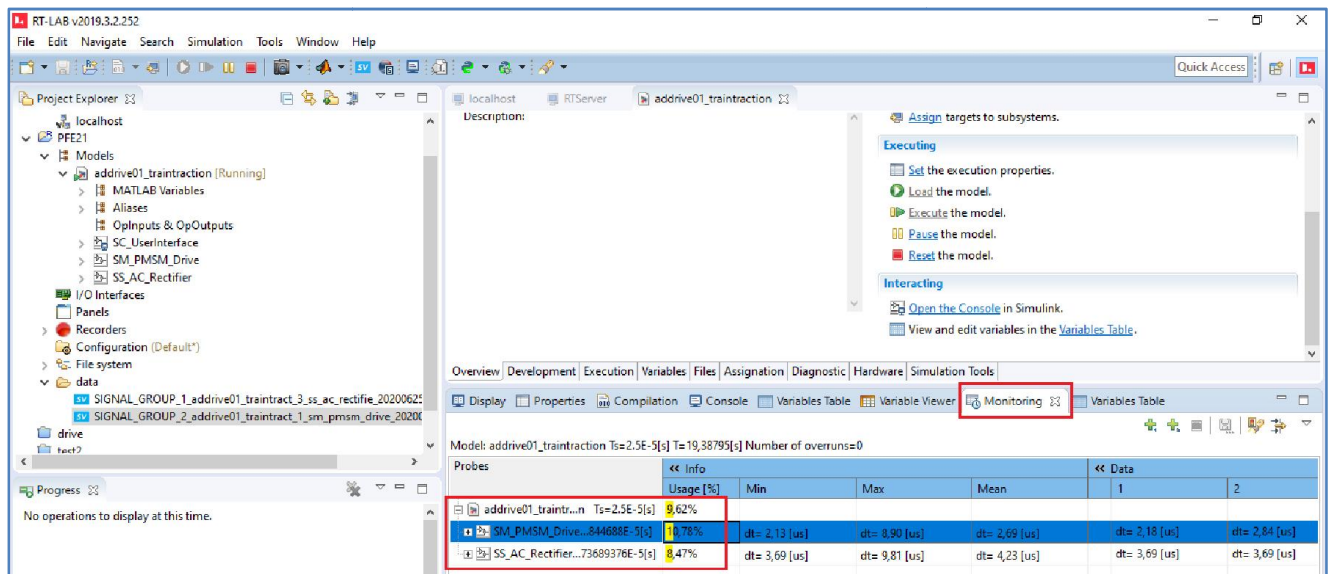


Figure (4.12-a): données de simulation en temps réel avec activation du mode XHP

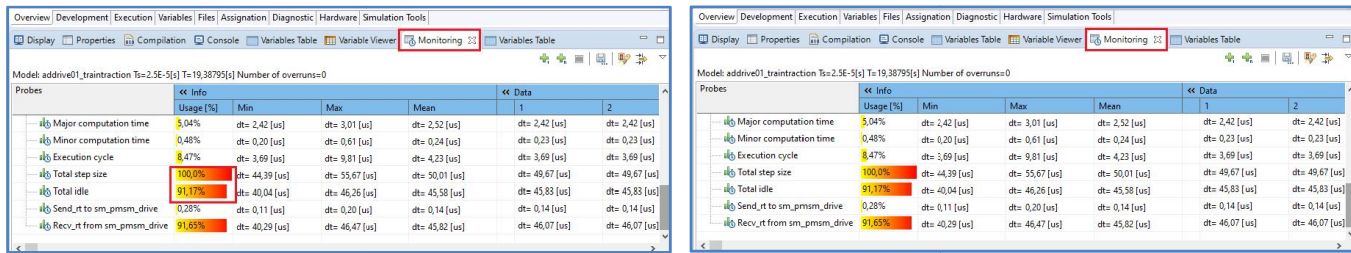


Figure (4.12-b): données de simulation en temps réel avec (XHP activé) pour les deux sous systèmes

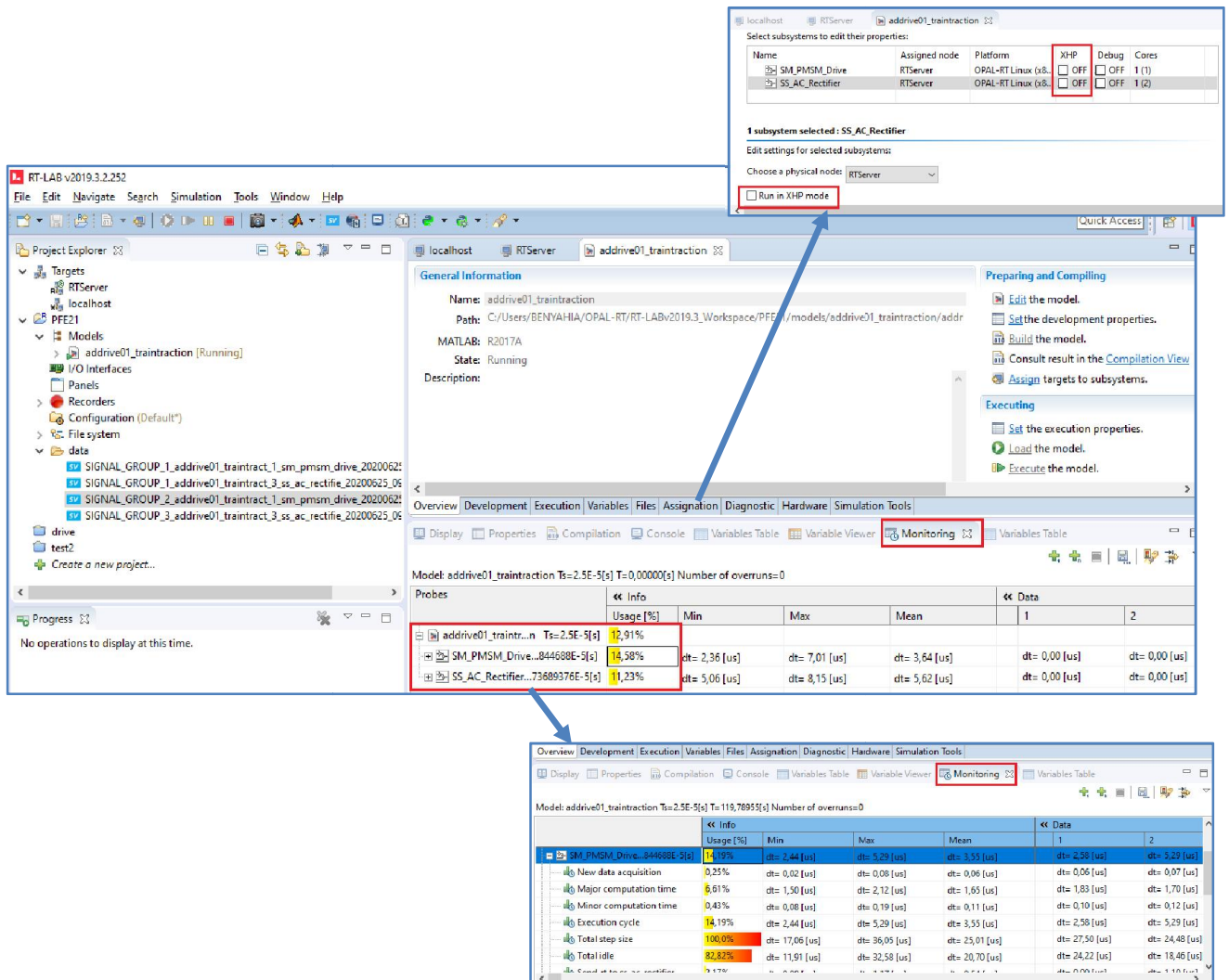


Figure (4.13): données de simulation en temps réel sans mode XHP

4.4.2 Résultats de simulation

Après avoir vérifié le déroulement de notre simulation en temps réel, on a choisi quelques paramètres et variables des différents sous systèmes à surveiller (monitoring), afficher et enregistrer afin de vérifier les réponses de notre modèle.

4.4.2.1 Résultats d'enregistrement avec *OpwriteFile (Matlab)*

Les enregistrements des résultats de simulation en temps réel avec Opwritefile sous forme de fichiers Matlab sont illustrés par les figures (4.14), (4.15) et (4.16).

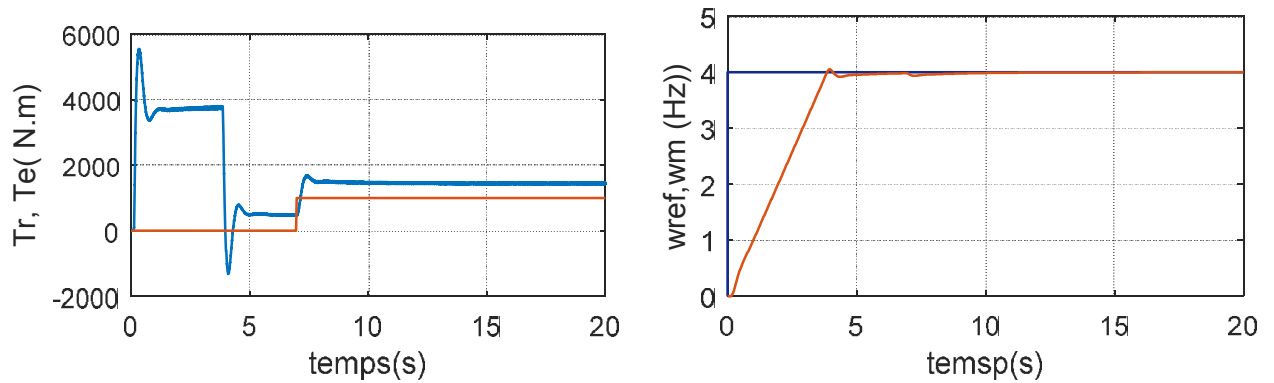


Figure (4.14) : Allures des couples (ΔT_{ref} , T_m) et vitesses (ω_{ref} , ω_m)

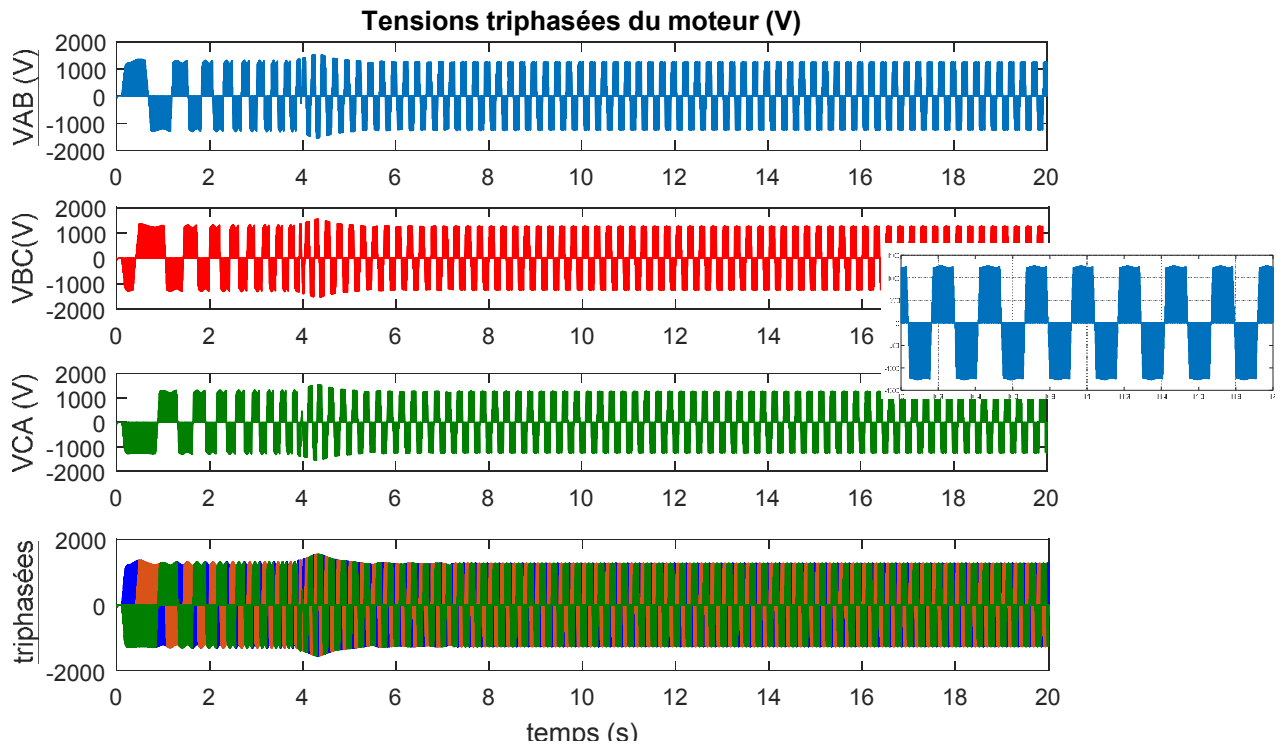


Figure (4.15) : Tensions triphasées du moteur

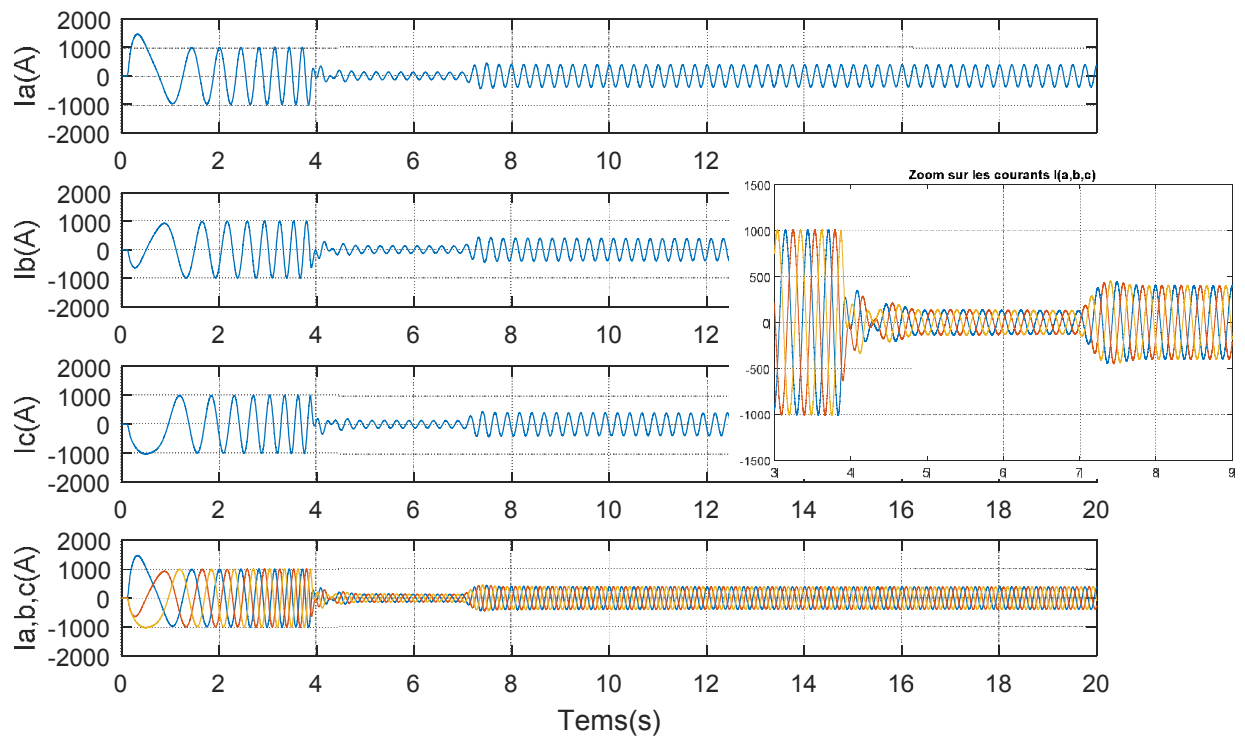


Figure (4.16) : Courants triphasés du moteur

4.4.2.2 Configuration de l'enregistrement sous ScopeView

L'application ScopeView de RT-LAB nous permet d'examiner les résultats graphiques ainsi que numériques enregistrés durant l'exécution de la simulation en temps réel.

Les signaux qu'on désire affichés seront sélectionnés pour chaque sous système, un menu de RT-LAB est destiné à configurer ses opérations juste après la compilation du modèle et avant son chargement sur le simulateur. Les figures ci-dessous montrent les différentes sélections.

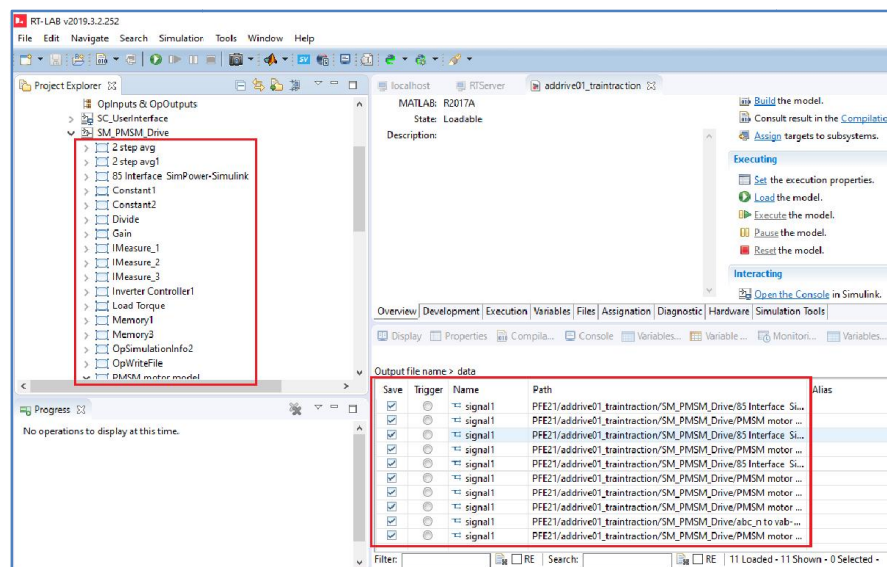


Figure (4.17) : Sélection et choix des signaux à enregistrer pour chaque sous système

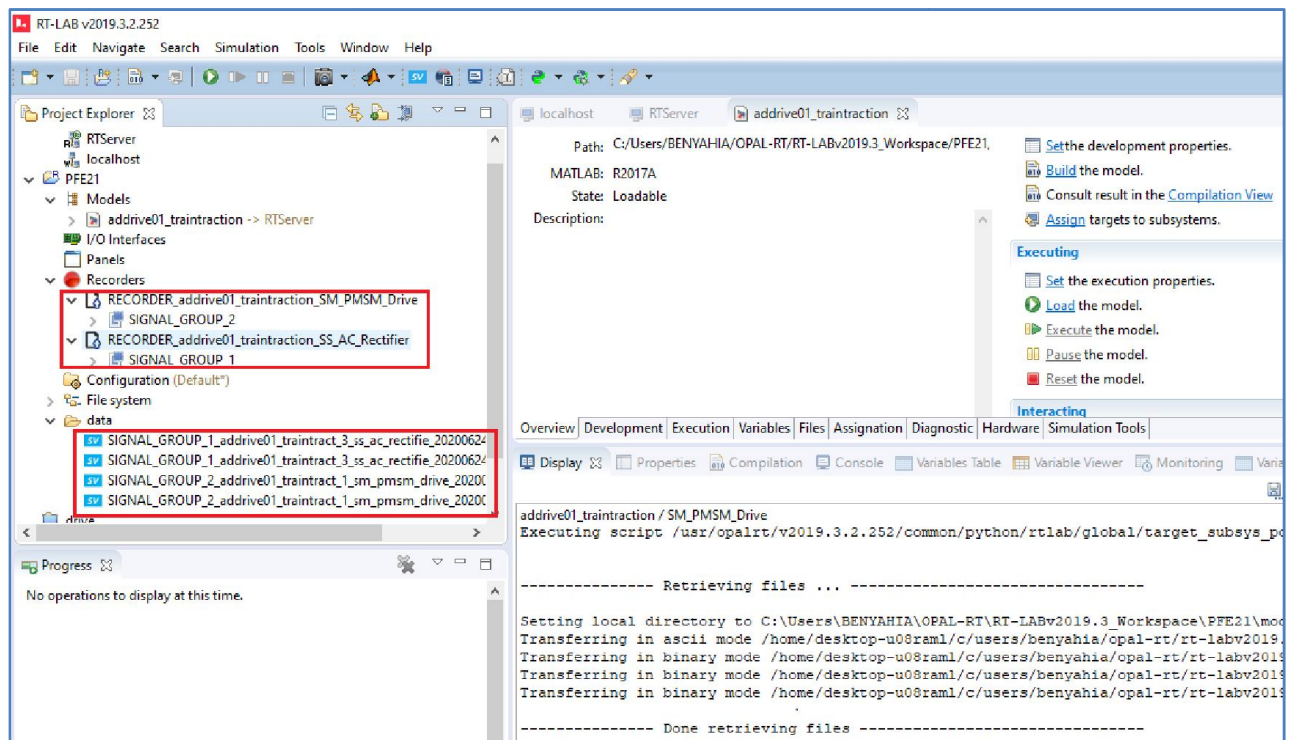


Figure (4.18) : Les groupes de signaux développés après exécution et enregistrement

Le lancement de ScopeView de puis RT-LAB est effectué dans notre cas après l'exécution et la réinitialisation de la simulation « reset ». (On peut même afficher les résultats durant l'exécution). Les figures (4.19) et (4.20) illustrent le lancement de Scopeview avec l'affichage des groupes de signaux sélectionnés.

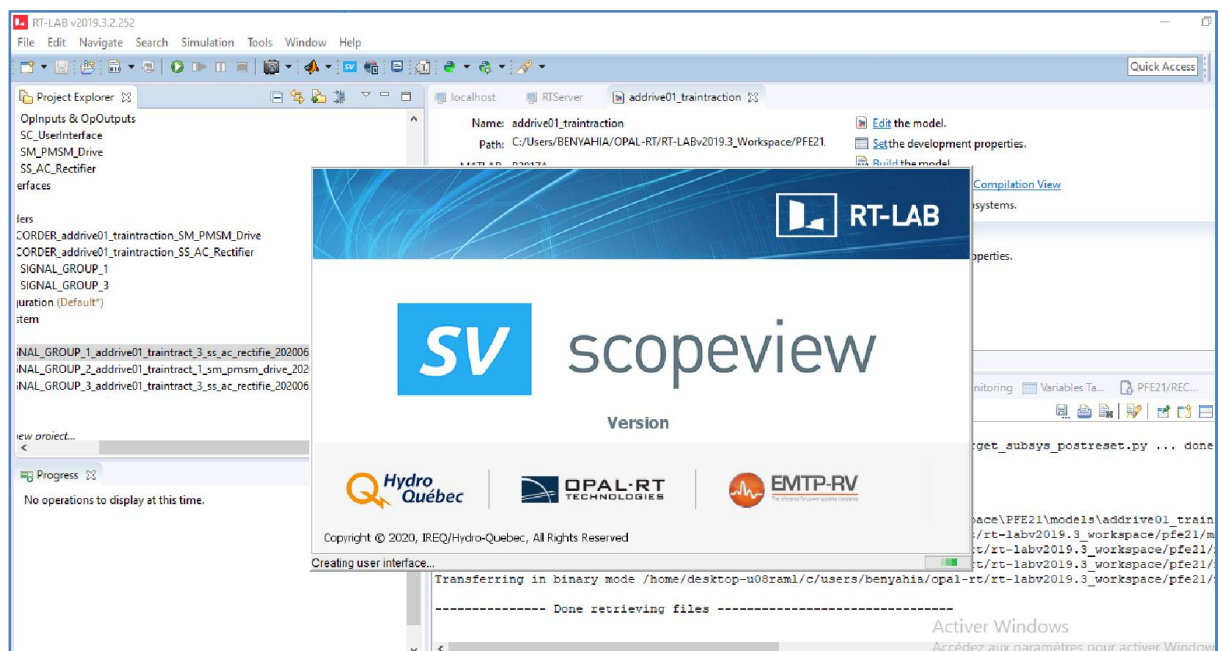


Figure (4.19) : Lancement de ScopeView

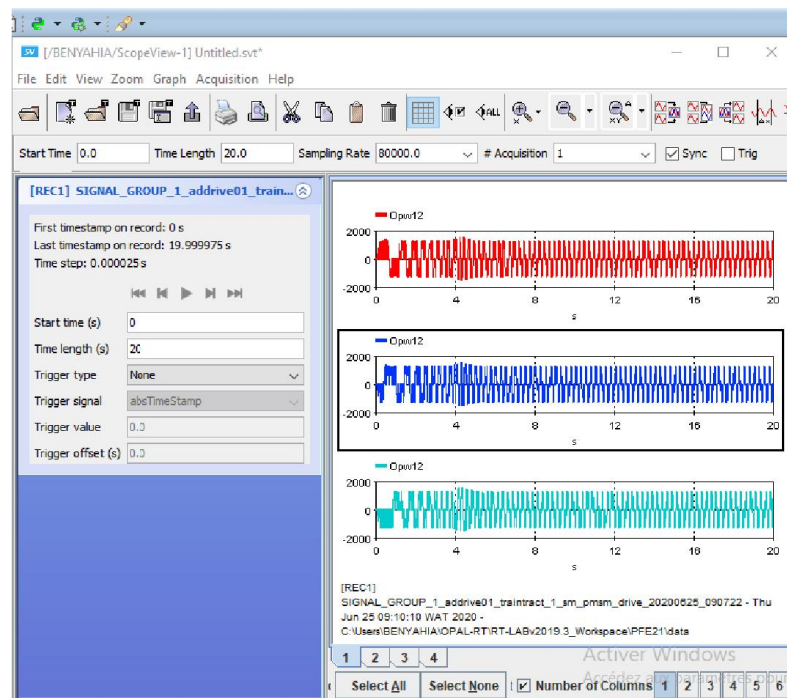
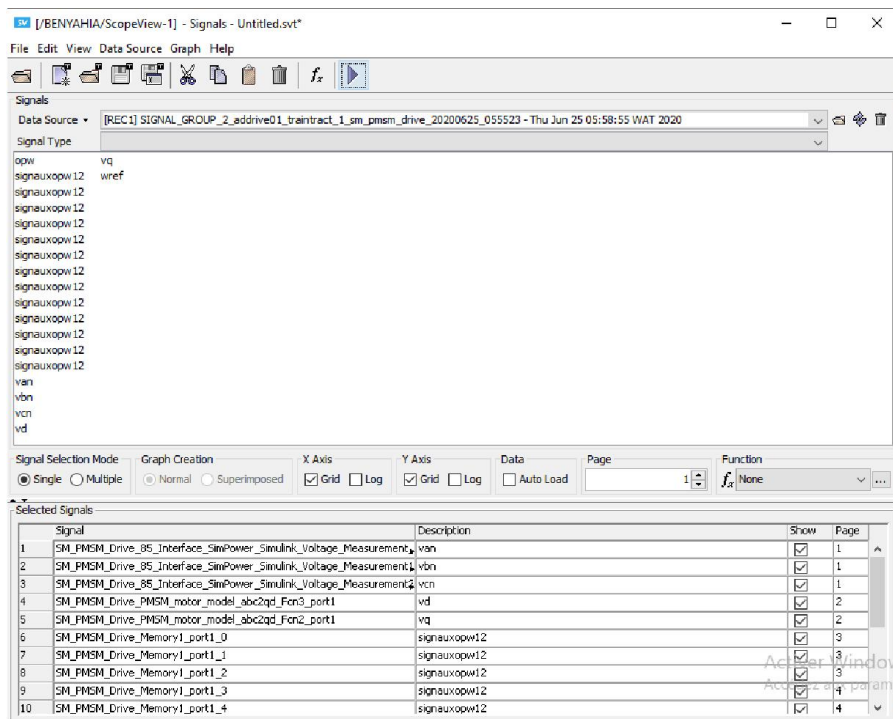


Figure (4.20) : l'interface d'affichage (graphique) de ScopeView

4.4.2.3 Enregistrement des résultats avec ScopeView

Les figures de (4.21) à (4.30) représentent les résultats de simulation enregistrés avec l'application ScopeView/RT-LAB.

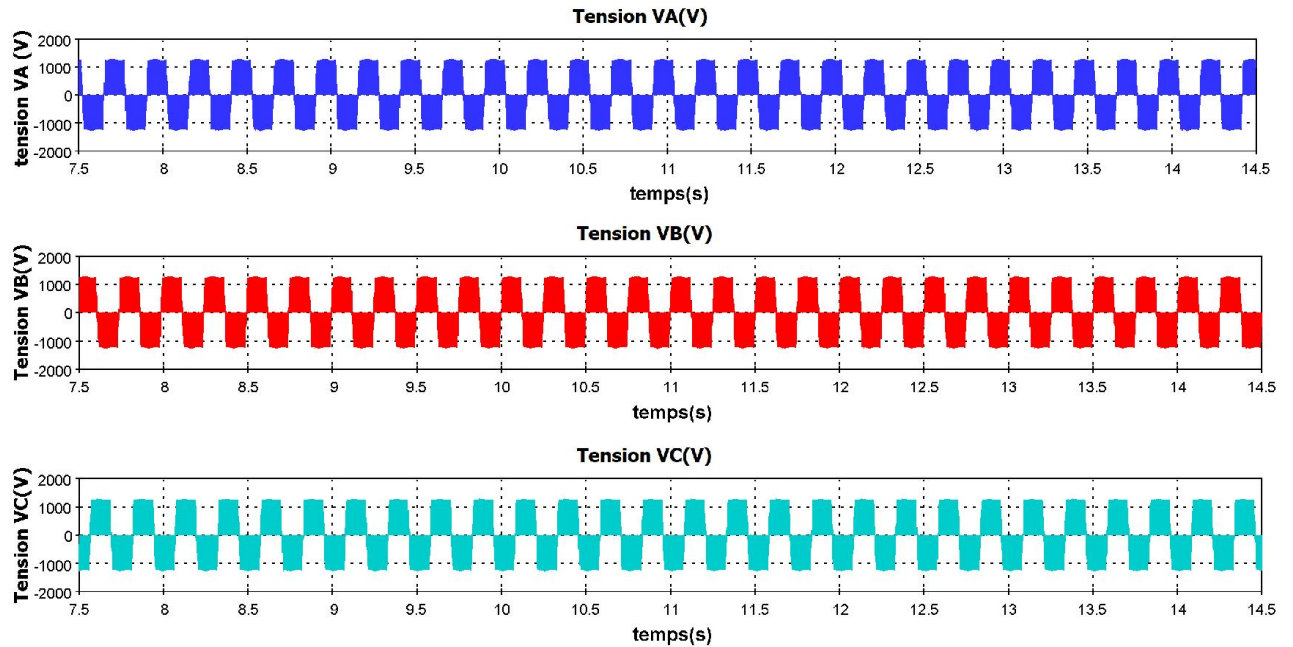


Figure (4.21-a) : Tensions triphasées du moteur – (enregistrement SV)

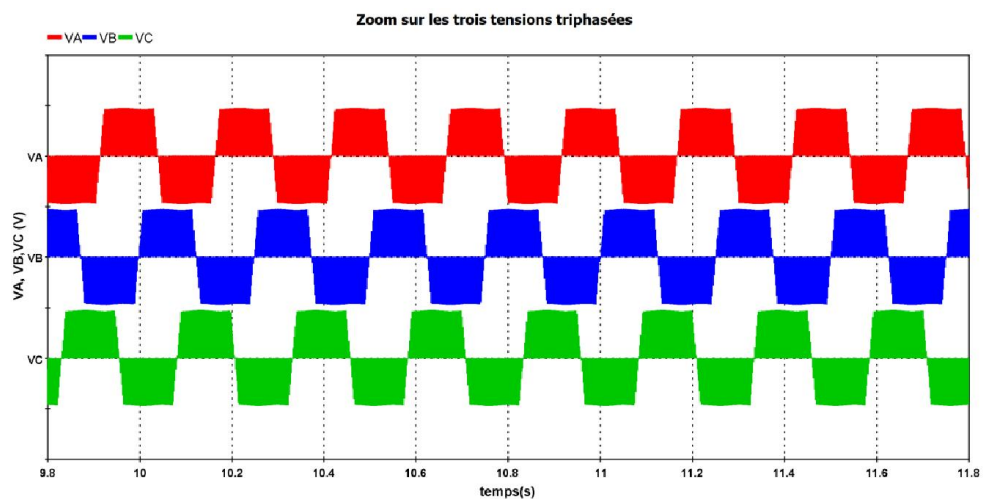


Figure (4.21-b) : Zoom sur les tensions triphasées du moteur – (enregistrement SV)

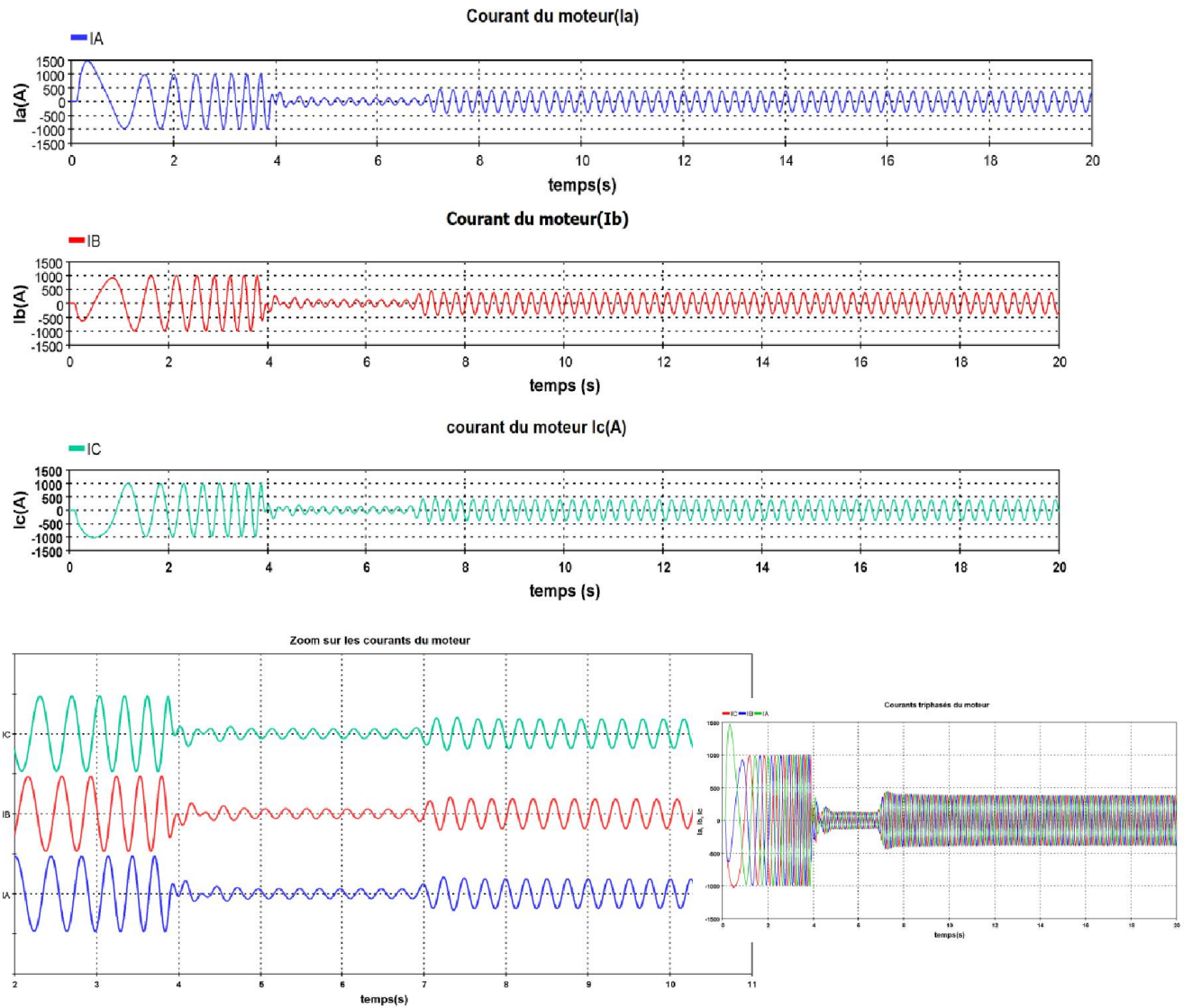


Figure (4.22) : Courants triphasés du moteur (enregistrement SV)

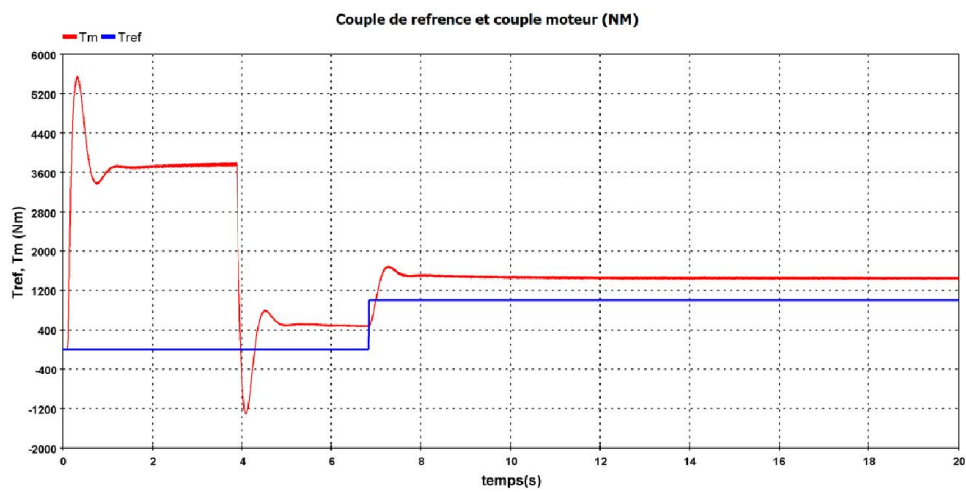


Figure (4.23) : Allures des couples (ΔT_{ref} , T_m - enregistrement SV)

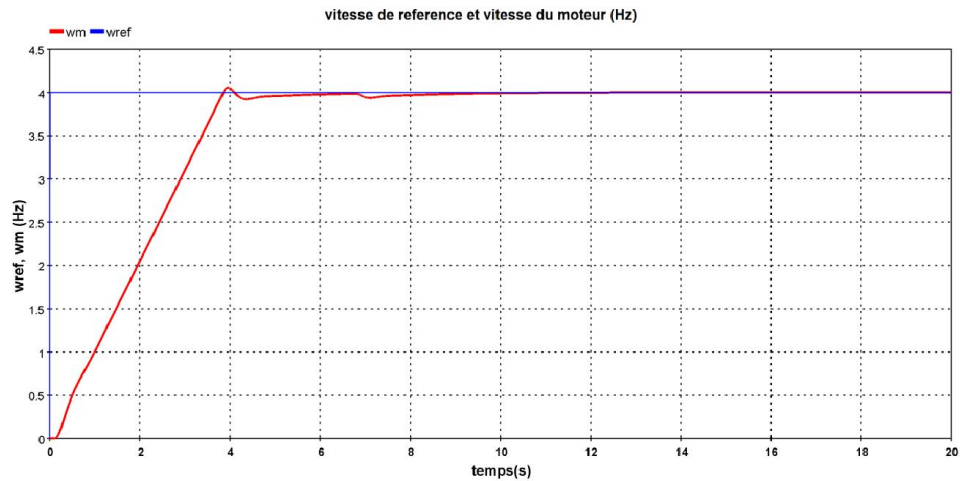


Figure (4.24) : Allures des vitesses (ω_{ref} , ω_m) (enregistrement SV)

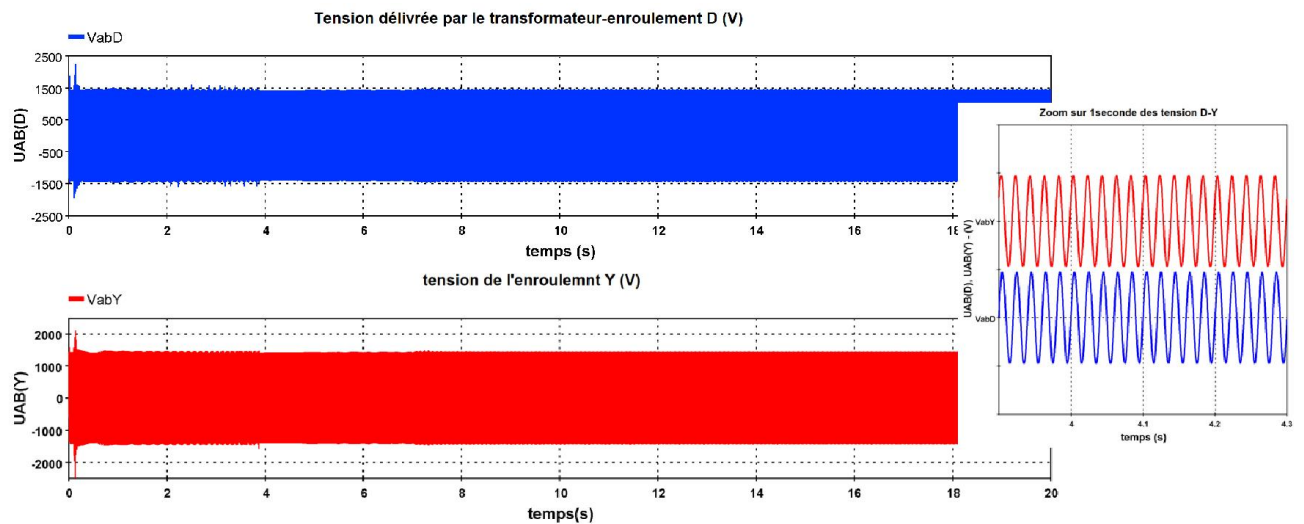


Figure (4.25) : Tensions aux secondaires du transformateur (V_D , V_Y) – (enregistrement SV)

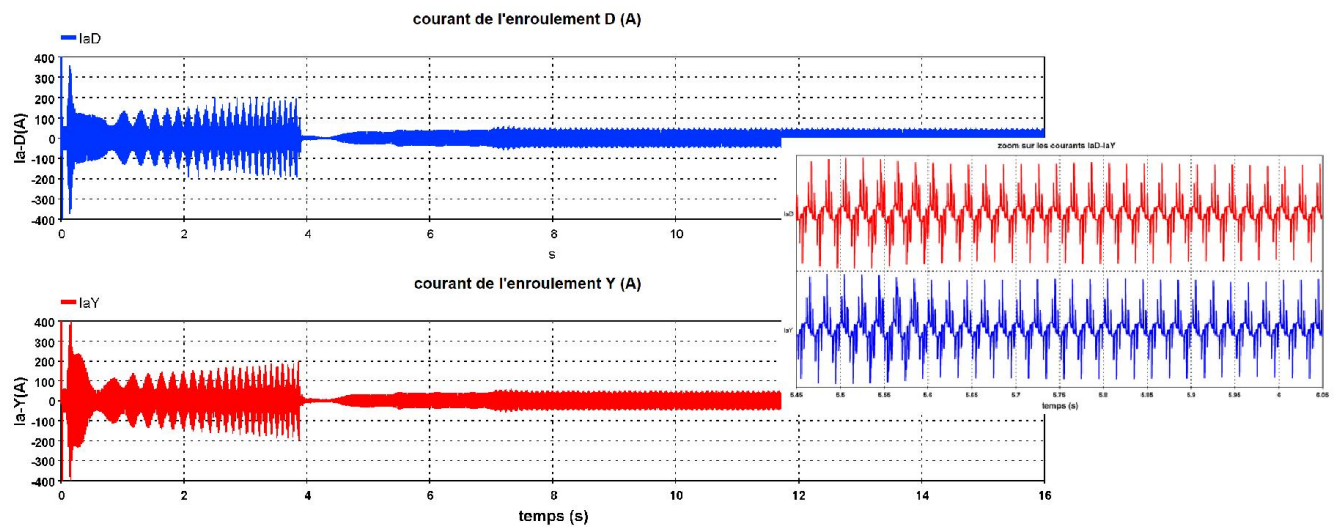


Figure (4.26) : Courants aux secondaires du transformateur (I_{aD} , I_{aY}) – (enregistrement SV)

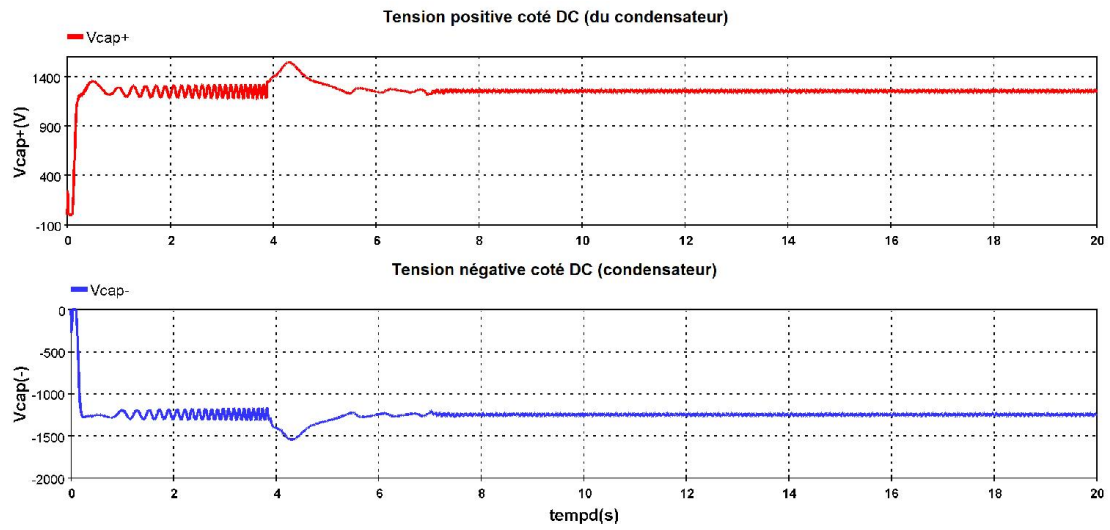


Figure (4.27) : Tensions du condensateur coté DC- (enregistrement SV)

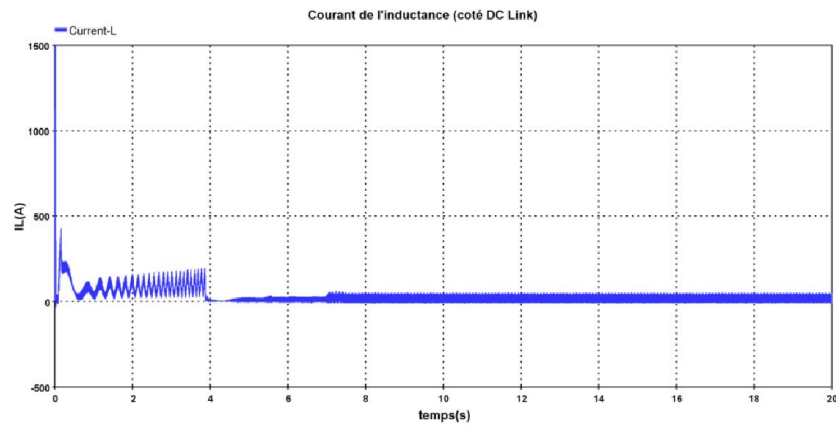


Figure (4.28) : Courant de l'inductance I_L (enregistrement SV)

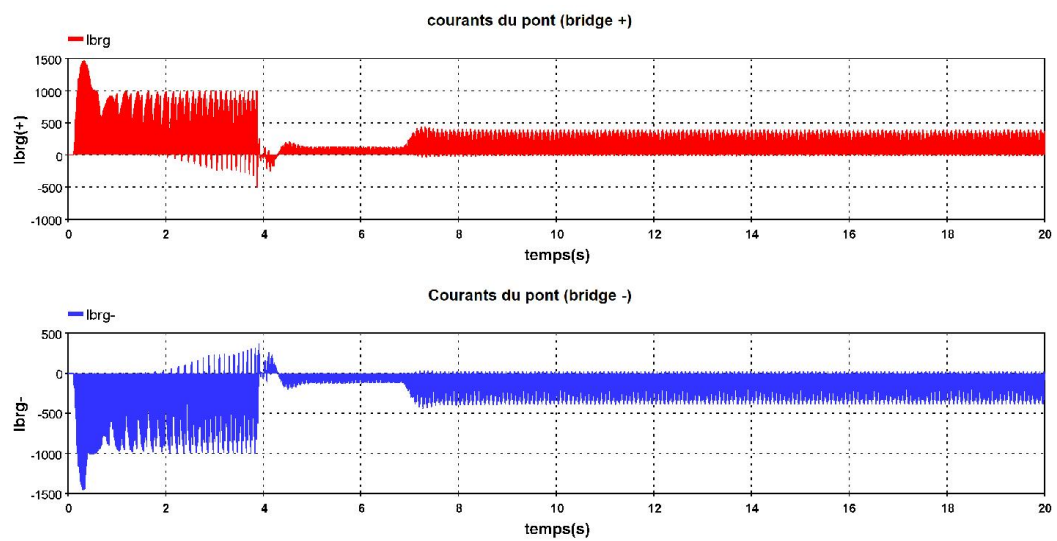


Figure (4.29) : Courant du pont onduleur I_{br} (enregistrement SV)

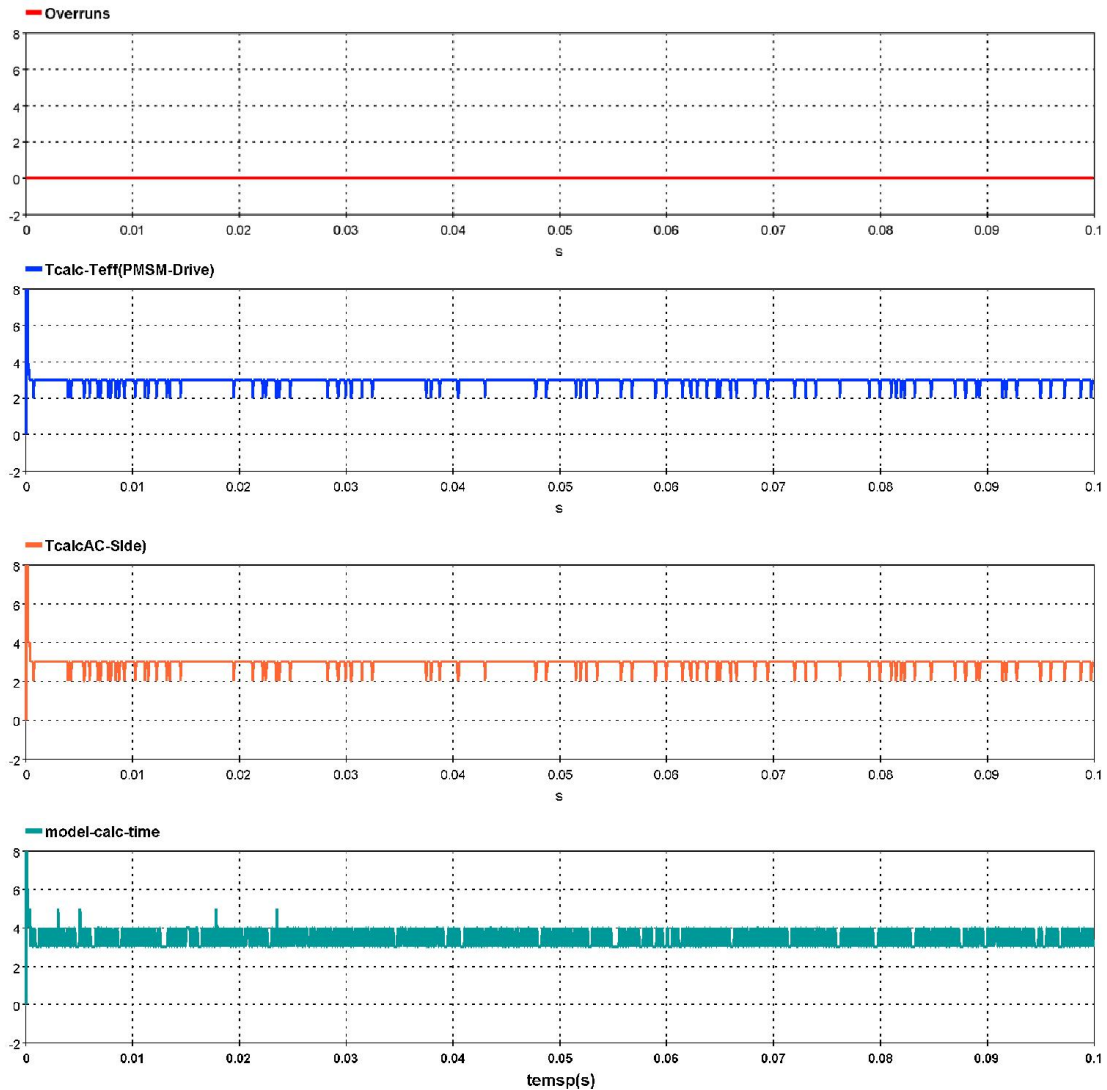


Figure (4.30) : Temps de calculs et dépassements (enregistrement SV)

4.5 Interprétations des simulations

D'Après ces résultats on remarque que la simulation en temps réel peut donner plus d'informations sur l'état de la simulation et de contrôler et surveiller les variations et le comportement du système (modèle) durant son exécution. Les signaux des réponses peuvent être enregistrés avec des précisions selon les prérequis bien choisis.

Le modèle simulé est à couple élevé et faible vitesse. La réponse lente remarquée est due à l'inertie du moteur et au contrôleur. Le contrôleur doit s'assurer que le bus DC est correctement chargé. Le moteur est démarré en charge, il développe alors un couple moteur important pour accélérer le train (courants et couple élevés dans les oscillogrammes). Après un transitoire de (4s) et lorsque la vitesse de consigne est atteinte, le couple et les courants

Diminuent, le moteur développe alors un couple moteur constant de (460 Nm). Ensuite, on a appliqué un couple de charge supplémentaire de l'ordre de (1000 Nm à $t = 7\text{s}$), la vitesse maintien toujours la valeur de consigne, vu quelle est contrôlée par un contrôleur assez complexe, avec des éléments de « feedforward » à anticipation [5].

4.6 Conclusion

On a présenté dans ce chapitre un exemple simulé avec RT-LAB sur le simulateur OP4510, concernant l'entraînement d'un moteur synchrone à aimants permanent (Train Traction DRIVE).

On s'est intéressé plus à la configuration, la préparation des modèles et aux préréglages nécessaires. Car cette phase est indispensable et demande des connaissances avancées, la vérification et la personnalisation des outils RT-LAB (Opcomm, Opctrl, Opwritefile, EFPGA, eHS, etc.), puisqu'on doit synchroniser deux plates formes Matlab/Simulink et RT-LAB.

On a montré que le simulateur en temps réel est capable d'effectuer une simulation en temps réel de dispositifs de propulsion complexes de traction électrique ici un train. La simulation en temps réel d'un entraînement par onduleur GTO à trois niveaux d'une machine synchrone à aimants permanents basé sur des convertisseurs à forts courants a été démontrée avec des résultats de simulation en temps réel. Le simulateur convient également à la hiérarchie de contrôle de niveau supérieur.

On a entamé plusieurs essais et réglages jusqu'à l'obtention des modèles convenables et qui fonctionnent sans erreurs (fautes) sur le simulateur OP4510. Les enregistrements graphiques nous ont permet de vérifier, comparer et discuter les résultats obtenus.

CONCLUSION GENERALE

CONCLUSION GÉNÉRALE

Les délais de conception réduits, de prototypage et d'essai de systèmes mécaniques et électriques complexes sont devenus utiles grâce à la proposition des systèmes de simulation à haute performance qui sont ouverts et faciles à utiliser, en utilisant les technologies du marché, optimisées pour le calcul parallèle ; rendre la simulation temps réel efficace, précise et accessible à tous. « La simulation temps réel est un outil indispensable pour les ingénieurs concepteurs. »

La motivation de ce projet est d'établir un support explicatif de la méthodologie, et du déroulement des étapes de simulation en temps réel sous la plateforme OPAL/RT-LAB. avec le simulateur OP4510. Ce support est renforcé par des essais sur des exemples bien choisis.

L'exemple simulé : « Train Traction-Drive », représente un entraînement de moteur synchrone à aimants permanents (PMSM) par un onduleur à GTO triphasé (3level-TSB Inverter), l'alimentation du bus continu est assuré par deux redresseurs à thyristors contrôlés.

La simulation en temps réel nous a permis d'implémenter les deux sous systèmes avec la plate-forme de simulation en temps réel RT-LAB d'Opal-RT Technologies à l'aide d'un ensemble de blocs Simulink. La compilation et la simulation globales du modèle sont entièrement automatisées par RT-LAB.

Puisque les deux sous systèmes emploient chacun des contrôleurs et calculateurs avec des temps d'échantillonnage différents (T_{s1} , T_{s2}), La compilation du modèle nécessite l'attribution et l'assignation de deux cœurs (CPU) du simulateur. L'activation du mode « XHP » est importante dans ce cas pour plus de précision et de rapidité de calcul.

La fonction « monitoring » intégrée nous a permis de surveiller les performances du modèle en affichant les temps de calcul de la simulation, les dépassements et les variations pour chaque sous système lors de son exécution en temps réel.

Le menu de la plate forme logicielle RT-LAB nous a offert les possibilités de superviser et perfectionner des paramètres et des variables « online » durant le déroulement de la simulation, jusqu'à l'enregistrement des résultats obtenus.

Les enregistrements ont été effectués soit par le bloc intégré « OpWritefile » soit par l'application « Scopeiew ».

À la fin, nous souhaitons que ce travail soit utile et compréhensible, car c'est l'œuvre d'un grand travail de traduction et d'apprentissage sur la manipulation du simulateur. Et nous espérons que d'autres fonctions seront activées sur ce simulateur afin de permettre à réaliser de futurs travaux sur la base des modèles « HIL ».

REFERENCES

BIBLIOGRAPHIQUES

Références bibliographiques

- [01] Chaudron, Jean-Baptiste, «Architecture de Simulation Distribuée Temps-Réel», Thèses en informatique, Toulouse, Université de Toulouse, 267p. 2012.
- [02] Manuels et Documentations OPAL-RTLAB technologies/OP4510. 2018, 2020
- [03] O. Mohammed and N. Abed, « Real-time simulation of electric machine drives with hardware-in-the-loop», COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 27(4), pp. 929-938, 2008.
- [04] C. Dufour, J. Mahseredjian, J. Belanger, « A Combined State-Space Nodal Method for the Simulation of Power System Transients », IEEE Transactions on Power Delivery, (ISSN 0885-8977), pp. 928-935, Vol. 26, no. 2, April 2011
- [05] Christian Dufour, Guillaume Dumur, Jean-Nicolas Paquin, Jean Bélanger « A Multi-Core PC-based Simulator for the Hardware-In-the-Loop Testing of Modern Train and Ship Traction Systems», IEEE Xplore: 30 September 2008, Opal-RT Technologies 1751 Richardson, suite 2525, Montreal, Canada.
- [06] <http://new.abb.com/drives>
- [07] Christian Dufour, « Highly stable rotating machine models using the state-space-nodal real-time solver », Opal-RT Technologies Montréal, Québec, Canada, COMPENG conference, Firenze, Italy. 2018.
- [08] Christian Dufour, Simon Abourida, «Hardware-In-the-Loop Simulation of Power Drives with RT-LAB», Conference: Power Electronics and Drives Systems, 2005. Volume: 2, Publisher: IEEE, IEEE Xplore: 18 April 2006.
- [09] Christian Dufour, Sébastien Cense, Vahid Jalili-Marandi, Jean Bélanger, « Review of state-of-the-art solver solutions for HIL simulation of power systems, power electronic and motor drives », EPE'13 ECCE Europe conference, September 3-5, 2013, Lille, France

ANNEXES

% init_addrive01_traintraction

%ParametersInitialization.m %

%Input Transformer

%Link Filter

Lzk=4e-3;

Czk=12e-3;

wn=1/sqrt(Lzk*Czk);

%if

%Lzk=0.4e-3;

%Czk=50e-3;

%wn=1/sqrt(Lzk*Czk);

%Output Transformer

Rta=0.01;

Lta=0.5e-3;

%Transrapid

Rl=0.1;

Ld=10e-3;

Lq=10e-3;

Psim=2.5;

J=500;

p=1;

%Rectifier

%Sampling Time of the rectifier controller

Tsg=500e-6;

%Rectifier Current Controller

Tgii=10e-3;

Kgip= 2.2048;

Kgii=220.4754;

Kgg=1.3780;

%Rectifier Voltage Regulator

Tgui=100e-3;

Kgup= 0.4800;

Kgui=4.800;

%Inverter

%Inverter Sampling Time

Tsw=2e-3;

%Inverter Current Controller

Kwip=1.6667;

Twii=0.0955;

Kwii=5.8201;

%Inverter Speed Controller

```
Twini=2;
Kwnp=1500;
Kwni=500;
```

```
%disp('overriden by dufour')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% 6-pulses thyristor rectifiers snubbers %%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Original Values
```

```
Thy_RsY=500;
Thy_CsY=0.1e-6;
Thy_RsD=500;
Thy_CsD=0.1e-6; %Original
```

```
% Modified Values
```

```
if 1==1
Thy_RsY=20;
Thy_CsY=10e-6;
Thy_RsD=20;
Thy_CsD=10e-6;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Xfo parameters %%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
T_Lpu=0.1103;
T_Lpu_pri=0.0061;
T_Rpu =0.002;
T_Rpu_pri=0.002;
T_Pbase=1.5e6;
T_fbase=50;
T_Vpri =20000;
T_Vsec =1000;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Decoupling Transformer snubbers %%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Transfo Y
```

```
T_Rsnu=10;
T_Csnu=10e-6;
T_Lsnu=(1/2/pi/50)^2/T_Csnu*0;
% transfo Delta
T_Rsnu2=10;
T_Csnu2=10e-6;
T_Lsnu2=(1/2/pi/50)^2/T_Csnu2*0;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%  
%% Shunt snubbers between Thyristors and filter inductors %%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%  
  
LC_Rsnu=5000;  
LC_Csnu=10e-6*inf;  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%% simulation info %%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
sys=bdroot;  
Ts=25e-6;  
Ts2=2*Ts;  
dec=1;
```